DNSSEC 機能確認手順書 Ver. 1.2

株式会社日本レジストリサービス
http://日本レジストリサービス.jp/
http://jprs.co.jp/
2010/01/25 Ver. 1.0（初版）
2010/01/26 Ver. 1.1
2010/07/21 Ver. 1.2

目次

I. はじめに....1
1. 本資料の目的....1
2. 本資料の構成....1
3. 確認項目の記述項目の選択基準....1
3.1. 関連 RFC からの対象項目の選出....1
3.2. 選択基準の策定....2
II. 環境....4
1. 本資料で用いた検証環境....4
2. 検証の条件....5
III. トラブルシューティングのシナリオ....6
シナリオ 1. 権威サーバへの問合せに常に失敗する....6
シナリオ 2. 権威サーバへの問合せに時々失敗する....8
シナリオ 3. 権威サーバへの問合せが遅い....9
シナリオ 4. 権威サーバへの問合せの結果が正しくない....10
シナリオ 5. フルリゾルバへの問合せに失敗する....13
シナリオ 6. フルリゾルバへの問合せに時々失敗する....15
シナリオ 7. フルリゾルバへの問合せが遅い....16
シナリオ 8. フルリゾルバが問合せの検証に失敗している....17
シナリオ 9. フルリゾルバへの問合せの結果が正しくない....20
IV. 確認項目....21
1. 権威サーバ側....21
確認項目 A-27. RRSIG レコードの有効期間終了フィールドが現在時刻より後であること....21
確認項目 A-28. RRSIG レコードの有効期間開始フィールドが現在時刻より前であること....21
確認項目 A-47. DS レコードのアルゴリズムは対応する DNSKEY レコードのものに一致するべき....24
確認項目 A-49. DS レコードのダイジェストは対応する DNSKEY レコードの鍵のハッシュであるべき....24
確認項目 A-50. DS レコードに対応する DNSKEY レコードはゾーン鍵であるべき....24
確認項目 A-58. 署名に使用した鍵がゾーン頂点の DNSKEY レコードに含まれていること....29
確認項目 A-61. 署名付きゾーンには SEP である DNSKEY RR(KSK 公開鍵情報)が最低 1

つ頂点になければならない 32
確認項目 A-68. 署名付きゾーン頂点の DNSKEY と親側の委任点にある DS が示すアルゴリズムの確認 35
確認項目 A-76. 子ゾーンが署名付きゾーンの場合、委任点に DS レコードが存在すべき 35
確認項目 A-78. DS は子ゾーンの頂点にあってはならない 40
確認項目 A-79. DS レコードは子ゾーン頂点の DNSKEY レコードを参照すべき 43
確認項目 A-80. 署名付きの子ゾーン頂点にある DNSKEY レコードは、対応する DS レコードと同じ秘密鍵で署名されるべき 43
確認項目 A-81. 子ゾーンの DS の TTL は、委任 NS の TTL と一致すべき 49
確認項目 A-85. セキュリティ対応権威サーバは EDNS0 による UDP 通信が可能であること 54
確認項目 A-86. セキュリティ対応権威サーバは 1220 バイトの UDP メッセージをサポートしていること 54
確認項目 A-87. セキュリティ対応権威サーバは 4000 バイトの UDP メッセージをサポートすべき 54
確認項目 A-95. セキュリティ対応権威サーバは DO=1 の問い合わせに応答する場合、RRSIG レコードが応答に含まれることの確認 63
確認項目 A-115. セキュリティ対応権威サーバは委任点の参照を応答する場合、DS とその RRSIG レコードが応答の権威部に含まれることの確認 66
確認項目 A-229. ゾーンの頂点に NSEC3PARAM レコードが存在すること。 70
確認項目 A-246. 非 opt-out 運用のゾーンで opt-out なしの NSEC3、またそれに対する RRSIG が返却されることの確認 73
確認項目 A-248. opt-out 運用のゾーンで opt-out の NSEC3 レコードが返却されることの確認 78
2. フルリゾルバ側 83
確認項目 F-2. DNSSEC 対応フルリゾルバの利用による AD ビットの確認 83
確認項目 F-27. RRSIG レコードの有効期間終了フィールドが現在時刻より後であること 86
確認項目 F-28. RRSIG レコードの有効期間開始フィールドが現在時刻より前であること 86
確認項目 F-47. DS レコードのアルゴリズムは対応する DNSKEY レコードのものに一致するべき 90
確認項目 F-49. DS レコードのダイジェストは対応する DNSKEY レコードの鍵のハッシュであるべき 90
確認項目 F-85. セキュリティ対応フルリゾルバは EDNS0 による UDP 通信が可能である

こと 95
確認項目 F-86. セキュリティ対応フルリゾルバは 1220 バイトの UDP メッセージをサポートしていること 95
確認項目 F-87. セキュリティ対応フルリゾルバは 4000 バイトの UDP メッセージをサポートすべき 95
確認項目 F-130. セキュリティ対応フルリゾルバは元の問い合わせの DO ビットに関わらず、再帰検索においては DO ビットを設定していること..... 105
確認項目 F-147. セキュリティ対応フルリゾルバの IP 層は IPv4 か v6 かに関わらず、フラグメントされた UDP パケットを正しく処理できなければならない..... 110
確認項目 F-154. セキュリティ対応フルリゾルバは少なくとも 1 つの信頼できる公開鍵または DS を設定に組み込める機能を持たねばならない..... 113
確認項目 F-194. 自分自身で署名され REVOKE bit の立った DNSKEY は Revoke される 118
確認項目 F-196. Revoke された DNSKEY は trust anchor として使用されない.... 118
確認項目 F-201. タイマー期限が過ぎたら新しい鍵は trust anchor に追加されること 118
確認項目 F-202. タイマー期限が来る前に新しい鍵は trust anchor に追加されていないこと 118
確認項目 F-229. ゾーンの頂点に NSEC3PARAM レコードが存在すること。 124
3. 共通項目 127
確認項目 共通-1. TCP の通信がブロックされていないことの確認 127
確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認 129
確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認 130
確認項目 共通-4. ping コマンドによる通信経路の MTU の確認 131
V. さいごに 133
1. 本手順書の扱いについて 133

Appendix A. DNSSEC 機能確認手順書の確認項目一覧

I. はじめに

1. 本資料の目的

本資料は、DNSSEC への普及期において、各 DNS サーバにおいて正しく DNSSEC サービスを提供できるようにするために必要な各種動作を確認するための手順書として作成している。そして、この目的を達成するため、実際のトラブルから想定される原因と、関連 RFC から抽出される各種の動作要求を「トラブルシューティングのシナリオ」として結び付けることを試みている。これらを通して、トラブルの解消を利用者が自力で行えるようになることを目的としている。

2. 本資料の構成

第 II 章「環境」では、本資料での動作検証のために構築した環境と、各ドメイン名のゾーン構成について説明している。

第 III 章「トラブルシューティングのシナリオ」では、DNSSEC を運用するにあたって想定できるトラブルを列挙し、それぞれのトラブルについて考えられる原因や、その確認・対処のための手順について記載している。

この章の使い方として、DNSSEC に関するトラブルに遭遇した場合、まず記載されているシナリオに該当するものがあるかをチェックし、該当するものがあれば、考えられる原因を把握し、確認するための手順をたどることによりトラブルを解消するという流れを想定している。

第 IV 章「確認項目」では、トラブルシューティングのシナリオから参照される確認手順の具体的な内容について、権威サーバ側、フルリゾルバ側に分けて記載している。各項目は第III章から参照されることを想定しているが、目次から対象とする確認項目を選び出し、確認する手順を知るために参照する、といった使い方も可能である。

3. 確認項目の記述項目の選択基準

3.1. 関連 RFC からの対象項目の選出

本手順書を作成するにあたり、DNSSEC におけるさまざまな運用上のトラブルを網羅できるように、記述項目を選ぶことを目標とした。

DNSSEC に関するトラブルは、運用ミスにより DNSSEC 関連の RFC に違反した状態となることによって発生する場合が多い。そこで、DNSSEC 関連 RFC から、運用上遵守しなければならない項目を選出し、対象項目一覧を作成した。対象とした DNSSEC 関連 RFC は次の通りである。

- RFC 4033 DNS Security Introduction and Requirements
- RFC 4034 Resource Records for the DNS Security Extensions
- RFC 4035 Protocol Modifications for the DNS Security Extensions
- RFC 5011 Automated Updates of DNS Security (DNSSEC) Trust Anchors
- RFC 5155 DNS Security (DNSSEC) Hashed Authenticated Denial of Existence

3. 2. 選択基準の策定

選出した対象項目一覧には、正しい運用によって準拠が保証されるものの他に、正しい実装を利用することによって準拠が保証されるものが相当数含まれており、実際に手順書を作成するには項目が多すぎる。

今回の手順書は DNSSEC の実装のためのものではなく運用のためのものであるため、項目一覧の中から運用のトラブルによって発生し得る項目を選択し、それらについての解決手順を記述するものとした。すなわち、運用ではなく権威サーバやリゾルバの実装によってのみ発生するものは、除外することとした。

しかしながら、運用か実装かの境界は必ずしも明確なものではないため、少なからずグレーゾーンが存在しており、明らかな形で明確に区分できるものとは言えないのが現状である。

そこで、本手順書では一定の選択基準を策定し、その基準に従ってこれを分類するものとした。具体的には以下の観点を各項目に適用し、項目を選択するものとした。

- 権威サーバあるいはフルリゾルバの設定ミスによって発生するもの
- ゾーンファイルの記述ミスによって発生するもの
- 経路上のネットワークの設定ミスによって発生するもの

「権威サーバあるいはフルリゾルバの設定ミスによって発生するもの」については、設定項目が多岐に渡るため、さらに次の仮定を置くものとした。

- 作業時点における最新の BIND バージョンである BIND 9.6.1-P1 の設定ファイル

named.conf の設定を想定する。

- これより古いバージョンのものは考慮しない。運用のバリエーションとして、設定可能な項目は最新のものにも含まれていると仮定する。
- 権威サーバあるいはフルリゾルバとしては BIND 9 以外のものも考慮するが、本手順書では、設定のバリエーションとしては BIND 9 が最も広いものと仮定し、他のソフトウェアは考慮しない。

上記の仮定の元、named.conf の設定項目について調査を行い、DNSSEC の運用トラブルを生じるものを抽出した。抽出された設定項目について手順の項目が影響を受けることを排除できないものは、記述対象とすることとした。

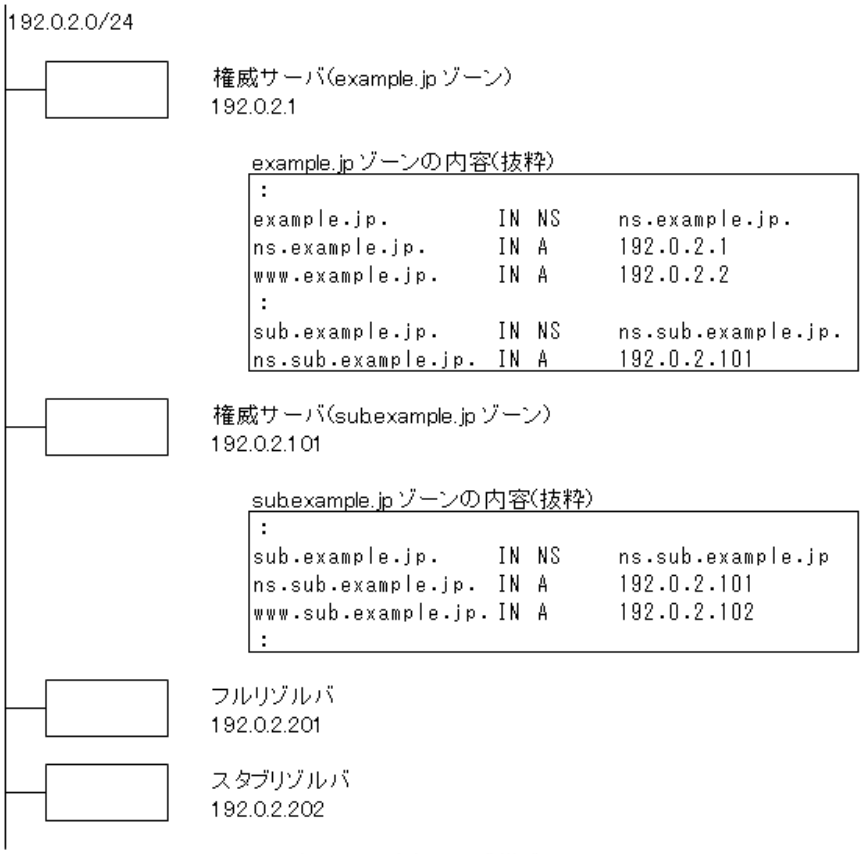
なお named.conf の設定項目の抽出については、設定によってログ出力等が異常になるもの、ゾーン転送が失敗する可能性があるもの、rndc コマンドや dynamic update 機能が正常に動作しなくなるものなど、DNSSEC に関連のないものは除外し、設定のミスにより関連 RFC に準拠しなくなる可能性を排除できないもののみを対象とすることとした。

さらにゾーンファイルの DNSSEC 署名は BIND 9 に含まれる dnssec-signzone コマンドで行うものとし「ゾーンファイルの記述ミスに発生するもの」については、署名後のゾーンファイルを直接編集するものを含めないものとした。

II. 環境

1. 本資料で用いた検証環境

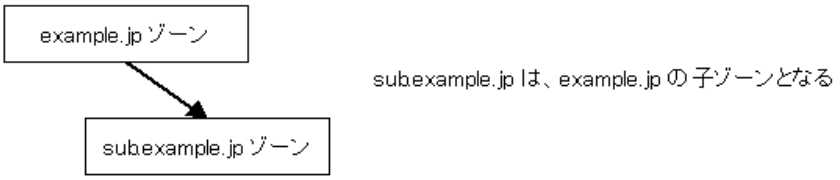
下記のようなテストネットワークを構築し、そこに以下の 3 台の DNS サーバと 1 台のテスト機を構築した。



ネットワークとサーバ構成

- 権威サーバ(example.jp):
  権威を持つゾーンとして example.jp ゾーンを定義し、いくつかのレコードを登録した。
  ゾーン情報を署名し、DNSSEC を有効とした。
  sub.example.jp への委任点を設定し、sub.example.jp の DS レコードを登録した。
- 権威サーバ(sub.example.jp):
  権威を持つゾーンとして sub.example.jp ゾーンを定義し、いくつかのレコードを登録した。

ゾーン情報を署名し、DNSSEC を有効とした。



構築したゾーンのツリー

- フルリゾルバ:
  特にゾーン情報は定義せず、再帰的名前解決を有効にした。
  DNSSEC を有効とした。

動作検証は、テスト機のスタブリゾルバのホストのコマンドプロンプトから、BIND 付属の dig コマンドを用いて各権威サーバ/フルリゾルバに対して実行することで行った。
また動作検証のパターンにより、それぞれの権威サーバのゾーン情報の署名あり/なしを切り替えたり、権威サーバ、フルリゾルバの DNSSEC の有効/無効を切り替えて検証を行った。

2. 検証の条件

動作検証に使用したソフトウェア、OS のバージョンは以下のとおり。

- 権威サーバ(example.jp):
  OS：CentOS release 5.4
  DNS サーバ：BIND 9.6.1-P1
- 権威サーバ(sub.example.jp):
  OS：CentOS release 5.4
  DNS サーバ：BIND 9.6.1-P1
- フルリゾルバ:
  OS：CentOS release 5.4
  DNS サーバ：BIND 9.6.1-P1、Unbound 1.4.0
- スタブリゾルバ:
  OS：CentOS release 5.4
  dig コマンド：BIND 9.6.1-P1 に付属のもの

III. トラブルシューティングのシナリオ

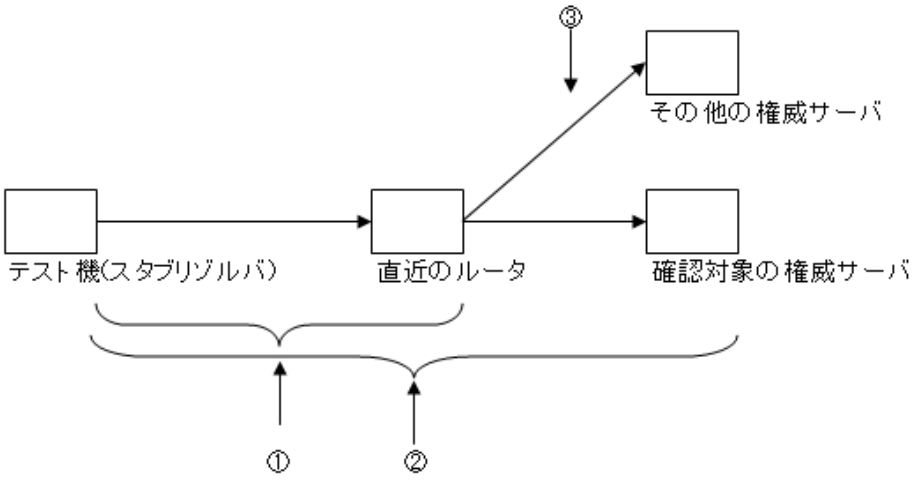
ここでは、DNSSEC を運用するにあたって想定できるトラブルを列挙し、そのトラブルの原因を調べるためにチェックをするべき各手順を示すシナリオを記載する。想定したトラブルの種類は次の通りである。

1. 権威サーバへの問合せに常に失敗する
2. 権威サーバへの問合せにときどき失敗する
3. 権威サーバへの問合せが遅い
4. 権威サーバへの問合せの結果が正しくない
5. フルリゾルバへの問合せに失敗する
6. フルリゾルバへの問合せにときどき失敗する
7. フルリゾルバへの問合せが遅い
8. フルリゾルバへの問合せの結果、検証に失敗している
9. フルリゾルバへの問合せの結果が正しくない

本手順書では、上記のトラブルに対して、対応するシナリオに沿って各チェック手順を実施し、原因の切り分け、分析を行うことを想定している。

シナリオ 1. 権威サーバへの問合せに常に失敗する

dig 等により権威サーバへの問合せを行うが、繰り返し様々な問合せを実施してみてもタイムアウト等によって常に問合せに失敗してしまうケースである。



図中の番号はこのシナリオにおいて問題が発生すると考えられる箇所であり、
下記の番号に対応している。この順に参照して問題解決を行うことを想定している。

1. 問合せを行っているテスト機にネットワーク的な問題はないか？

テスト機から直近のルータまでの接続性を確認する。直近のルータ(デフォルトルータ)のアドレスを引数として ping コマンドを実行し、結果を確認する。なお、ここでルータのアドレスは名前ではなく IPv4/IPv6 のアドレスを直接使い、名前解決を行わないようにする。

- Destination net unreachable あるいは Destination host unreachable が報告されているならば、テスト機のルート設定に問題がある。デフォルトルートが正しく設定されているかの確認を行う。
- Time out が発生する場合、直近のルータまでの物理リンクが正常でないか、あるいは ICMP がテスト機あるいは直近のルータでフィルタリングされている。物理ネットワークの接続性確認、およびテスト機および直近のルータのフィルタリング設定の確認を行う。

ここまでの確認項目に問題はないが、このシナリオが解決しない場合は、次に進む。

2. 問合せを行うテスト機と権威サーバとの間のネットワークに問題はないか？

テスト機から権威サーバまでのパケットの接続性を確認する。目標の権威サーバの IP アドレスを引数として ping コマンドを実行し結果を確認する。

- Destination net unreachable あるいは Destination host unreachable が報告されているならば、途中のルーティングに問題がある。対象の権威サーバの IP アドレスを再度確認の上、そのアドレスがテスト機と異なるネットワークに属するものであるならば、異なる権威サーバに対する接続性を確認し、問題の切り分けを行う。
- Time out が発生する場合、権威サーバが停止しているか、あるいは ICMP が権威サーバまでの経路途中でフィルタリングされている。異なる権威サーバに対する接続性を確認し、問題の切り分けを行う。
- 問題なく ping の Reply があれば、権威サーバまでのネットワーク到達性に問題はなく、また権威サーバ機自体も起動しているが、権威サーバプロセスが動作していないか、UDP port 53 に対するフィルタリングが行われていないかの確認を行う。

ここまでの確認項目に問題はないが、このシナリオが解決しない場合は、次に進む。

3. テスト機にインターネットからの 512 オクテットを超える DNS パケットが到達可能か？

確認対象の権威サーバとは異なる権威サーバであり、かつ DNSSEC 署名をすでに行っている権威サーバに対し(a.ns.se, …, j.ns.se 等が利用できる)、以下を参照して 512 オクテットを超えるような DNS 問い合わせを実行する。

確認項目 A-85. セキュリティ対応権威サーバは EDNS0 による UDP 通信が可能であること

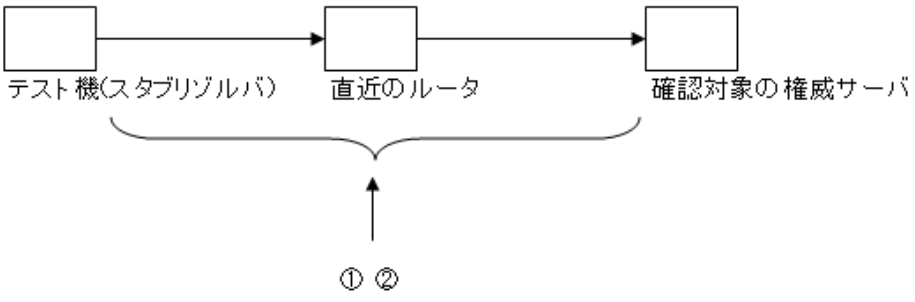
- TCP に切り替わり、TCP での通信に失敗しているようならば、TCP port 53 に対するフィルタリングが行われている可能性がある。上記確認項目を参照する。
- 権威サーバからの応答サイズが 1220 オクテットを越えるケースで失敗するならば、UDP のフラグメントが発生し、テスト機のネットワークで(あるいはインターネットで)、フラグメントのフィルタリングが行われている可能性がある。以下を参照する。

  確認項目 A-86. セキュリティ対応権威サーバは 1220 バイトの UDP メッセージをサポートしていること

  確認項目 A-87. セキュリティ対応権威サーバは 4000 バイトの UDP メッセージをサポートすべき
- 確認対象とは異なる権威サーバに対し、512 オクテットを超える DNS パケットが到達できる場合には、ネットワークの問題はないと考えられるので、上記確認項目を参照し、確認対象の権威サーバの設定を確認する。

シナリオ 2. 権威サーバへの問合せに時々失敗する

失敗する状況を確認する。失敗したものと同じ問合せを同一権威サーバに対して何度か行い、常に失敗しているか確認する。



図中の番号はこのシナリオにおいて問題が発生すると考えられる箇所であり、下記の番号に対応している。この順に参照して問題解決を行うことを想定している。

1. 問い合わせの内容が同じであっても、成功したり失敗したりする

DNSSEC や DNS の問題以前に、権威サーバまでの経路のネットワークの問題である可能性がある。

- シナリオ 1.を参照してネットワークの問題の切り分けを行い、権威サーバまでの安定したネットワークの確保を行う。

ここまでの確認項目に問題はないが、このシナリオが解決しない場合は、次に進む。

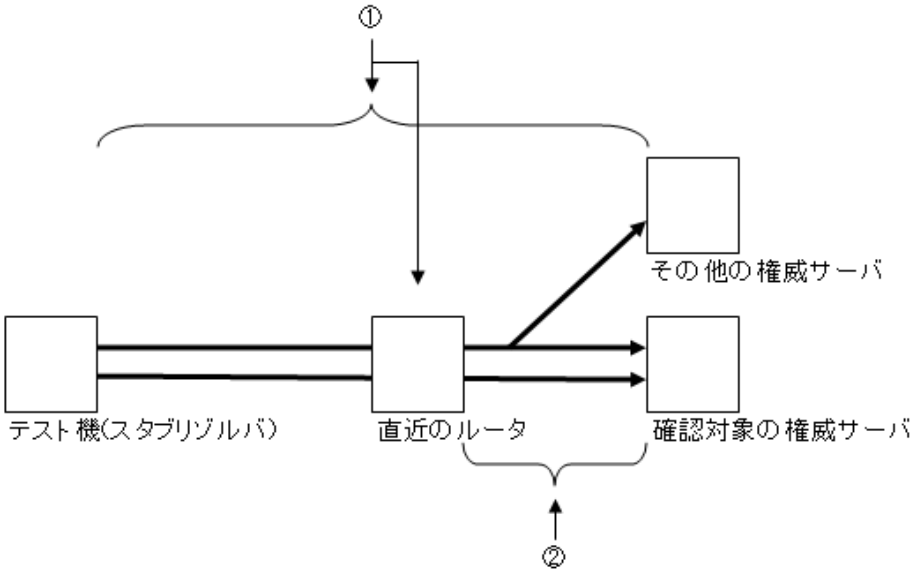
2. 問い合わせの内容により、必ず成功する(失敗する)ように見える

DNSSEC によって問い合わせに対する応答結果のサイズが増えた結果として、問い合わせに対し、あるサイズまでの応答は成功するが、あるサイズを超えるとパケットが落とされる等の問題が起きている可能性がある。

- シナリオ 1.や以下を参照し、EDNS0 通信が可能かどうかを確認する。
  確認項目 A-85. セキュリティ対応権威サーバは EDNS0 による UDP 通信が可能であること

シナリオ 3. 権威サーバへの問合せが遅い

確認対象の権威サーバだけでなく別の権威サーバへの問合せを試み、同様に遅くなるかどうかを確認する。



図中の番号はこのシナリオにおいて問題が発生すると考えられる箇所であり、
下記の番号に対応している。この順に参照して問題解決を行うことを想定している。

1. 別の権威サーバに対しても、同様に遅い

権威サーバ側ではなく、問い合わせを行うテスト機側のネットワークに問題がある可能性がある。

- 直近のルータや、テスト機を収容しているネットワークを調査し、ネットワーク障害が起きていないか調べる。
- シナリオ 1.を参照してネットワークの問題の切り分けを行い、権威サーバまでの安定したネットワークの確保を行う。

2. 別の権威サーバでは問題なく、確認対象の権威サーバのみ遅い

確認対象の権威サーバ側のみに問題が発生している可能性がある。

- 以下を参照し、遅くなっている原因を調査する。
  確認項目 A-85. セキュリティ対応権威サーバは EDNS0 による UDP 通信が可能であること

シナリオ 4. 権威サーバへの問合せの結果が正しくない

問合せの結果が想定した結果とならなかった場合、何が正しくないのかによって分類する。

1. DNSKEY レコードがない、あるいは想定した値ではない

権威サーバの設定において DNSSEC が有効になっていない、あるいはゾーンファイルの署名が行われていない、署名の有効期間が適切でない可能性がある。
以下の点を確認する。

- **確認項目 A-27. RRSIG レコードの有効期間終了フィールドが現在時刻より後であること**
- **確認項目 A-28. RRSIG レコードの有効期間開始フィールドが現在時刻より前であること**
- **確認項目 A-58. 署名に使用した鍵がゾーン頂点の DNSKEY レコードに含まれていること**
- **確認項目 A-61. 署名付きゾーンには SEP である DNSKEY RR(KSK 公開鍵情報)が最低 1 つ頂点になければならない**
- **確認項目 A-79. DS レコードは子ゾーン頂点の DNSKEY レコードを参照すべき**
- **確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認**
- **確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認**

2. RRSIG レコードがない、あるいは想定した値ではない

権威サーバの設定において DNSSEC が有効になっていない、あるいはゾーンファイルの署名が行われていない、署名の有効期間が適切でない可能性がある。

以下の点を確認する。

- **確認項目 A-27. RRSIG レコードの有効期間終了フィールドが現在時刻より後であること**
- **確認項目 A-28. RRSIG レコードの有効期間開始フィールドが現在時刻より前であること**
- **確認項目 A-76. 子ゾーンが署名付きゾーンの場合、委任点に DS レコードが存在すべき**
- **確認項目 A-95. セキュリティ対応権威サーバは DO=1 の問い合わせに応答する場合、RRSIG レコードが応答に含まれることの確認**
- **確認項目 A-115. セキュリティ対応権威サーバは委任点の参照を応答する場合、DS とその RRSIG レコードが応答の権威部に含まれることの確認**
- **確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認**
- **確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認**

3. NSEC レコードがない、あるいは想定した値ではない

権威サーバの設定において DNSSEC が有効になっていない、あるいはゾーンファイルの署名が行われていない、署名の有効期間が適切でない可能性がある。

また、DNSSEC の不在証明を NSEC レコード形式ではなく NSEC3 レコード形式で行っている可能性がある。

以下の点を確認する。

- **確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認**
- **確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認**

4. DS レコードがない、あるいは想定した値ではない

権威サーバの設定において DNSSEC が有効になっていない、あるいはゾーンファイルの署名が行われていない、署名の有効期間が適切でない可能性がある。

また親ゾーンの DS レコードと子ゾーンの DNSKEY レコード(KSK 公開鍵)の対応が正しくない可能性がある。

以下の点を確認する。

- 確認項目 A-47. DS レコードのアルゴリズムは対応する DNSKEY レコードのものに一致するべき
- 確認項目 A-76. 子ゾーンが署名付きゾーンの場合、委任点に DS レコードが存在すべき
- 確認項目 A-78. DS は子ゾーンの頂点にあってはならない
- 確認項目 A-79. DS レコードは子ゾーン頂点の DNSKEY レコードを参照すべき
- 確認項目 A-81. 子ゾーンの DS の TTL は、委任 NS の TTL と一致すべき

5. NSEC3PARAM レコードがない、あるいは想定した値ではない

権威サーバの設定において DNSSEC が有効になっていない、あるいはゾーンファイルの署名が行われていない、署名の有効期間が適切でない可能性がある。
また、DNSSEC の不在証明を NSEC3 レコード形式ではなく NSEC レコード形式で行っている可能性がある。
以下の点を確認する。

- 確認項目 A-229. ゾーンの頂点に NSEC3PARAM レコードが存在すること。
- 確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認
- 確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認

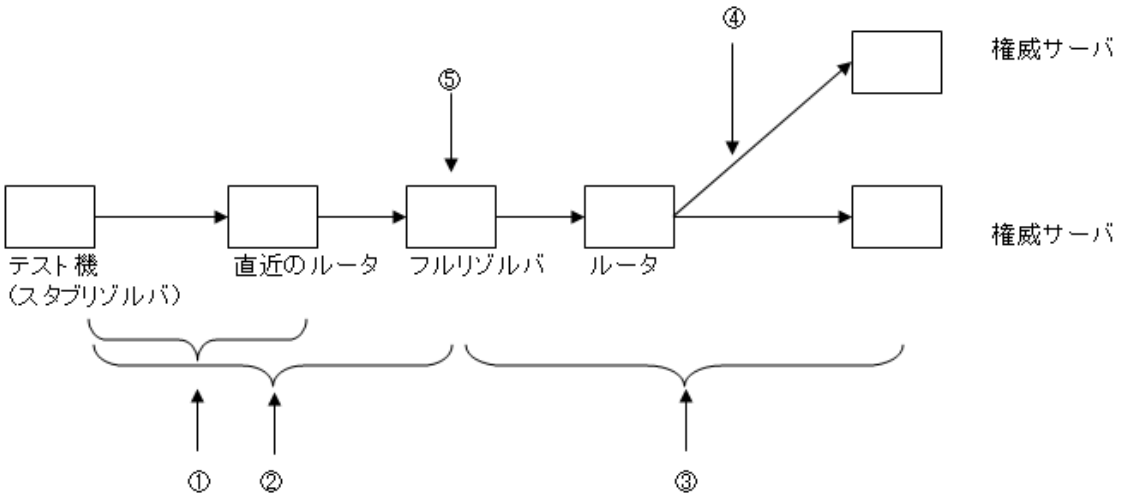
6. NSEC3 レコードがない、あるいは想定した値ではない

権威サーバの設定において DNSSEC が有効になっていない、あるいはゾーンファイルの署名が行われていない、署名の有効期間が適切でない可能性がある。
また、DNSSEC の不在証明を NSEC3 レコード形式ではなく NSEC レコード形式で行っている可能性がある。
以下の点を確認する。

- 確認項目 A-229. ゾーンの頂点に NSEC3PARAM レコードが存在すること。
- 確認項目 A-246. 非 opt-out 運用のゾーンで opt-out なしの NSEC3、またそれに対する RRSIG が返却されることの確認
- 確認項目 A-248. opt-out 運用のゾーンで opt-out の NSEC3 レコードが返却されることの確認
- 確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認
- 確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認

シナリオ 5. フルリゾルバへの問合せに失敗する

dig 等によりフルリゾルバへの問合せを行うが、繰り返し様々な問合せを実施してみてもタイムアウト等によって常に問合せに失敗してしまうケースである。



図中の番号はこのシナリオにおいて問題が発生すると考えられる箇所であり、
下記の番号に対応している。この順に参照して問題解決を行うことを想定している。

1. 問合せを行っているテスト機にネットワーク的な問題はないか？

テスト機から直近のルータまでの接続性を確認する。直近のルータ(デフォルトルータ)のアドレスを引数として ping コマンドを実行し、結果を確認する。なお、ここでルータのアドレスは名前ではなく IPv4/IPv6 のアドレスを直接使い、名前解決を行わないようにする。

- シナリオ 1.の 1.を参照し、同様のことをテスト機と直近のルータとの間で行う。

ここまでの確認項目に問題はないが、このシナリオが解決しない場合は、次に進む。

2. 問合せを行うテスト機とフルリゾルバとの間のネットワークに問題はないか？

テスト機からフルリゾルバまでのパケットの接続性を確認する。目標のフルリゾルバの IP アドレスを引数として ping コマンドを実行し結果を確認する。

- シナリオ 1.の 2.を参照し、同様のことをテスト機とフルリゾルバとの間で行う。

ここまでの確認項目に問題はないが、このシナリオが解決しない場合は、次に進む。

3. フルリゾルバのネットワーク、フルリゾルバと権威サーバとの間のネットワークに問

題はないか？

フルリゾルバを稼動させているサーバにログインが出来るのであれば、サーバにログインし、フルリゾルバとルータ、フルリゾルバと権威サーバとの間のネットワークの接続性を確認する。

- シナリオ 1.の 1.と 2.を参照し、同様のことをフルリゾルバと権威サーバとの間で行う。

ここまでの確認項目に問題はないが、このシナリオが解決しない場合は、次に進む。

4. フルリゾルバから権威サーバに対して、問い合わせは問題なく行えるか？

フルリゾルバを稼動させているサーバにログインが出来るのであれば、サーバにログインし、そのサーバからいくつかの権威サーバに対して dig コマンドを用いて問い合わせを行う。フルリゾルバと権威サーバとの間では問い合わせに問題がないことを確認する。

- シナリオ 1.の 3.を参照し、同様のことをフルリゾルバからいくつかの権威サーバとの間で行う。

ここまで確認できれば、フルリゾルバと権威サーバとの間ではネットワークに問題はなく、DNS の問い合わせも正常に行えていることになる。

ここまでの確認項目に問題はないが、このシナリオが解決しない場合は、次に進む。

5. フルリゾルバの設定に問題はないか？

フルリゾルバの設定が適切でない可能性がある。フルリゾルバが DNSSEC 対応として正しく設定されているかどうかを確認する。

- フルリゾルバは EDNS0 通信をサポートし、512 オクテットを超える udp パケットを正常に扱えること。また TCP への切り替わりが発生しても問題なく問い合わせを行えること。

  以下を参照する。

  確認項目 F-85. セキュリティ対応フルリゾルバは EDNS0 による UDP 通信が可能であること

  確認項目 F-86. セキュリティ対応フルリゾルバは 1220 バイトの UDP メッセージをサポートしていること

  確認項目 F-87. セキュリティ対応フルリゾルバは 4000 バイトの UDP メッセージをサポートすべき

  確認項目 F-130. セキュリティ対応フルリゾルバは元の問い合わせの DO ビットに関わらず、再帰検索においては DO ビットを設定していること

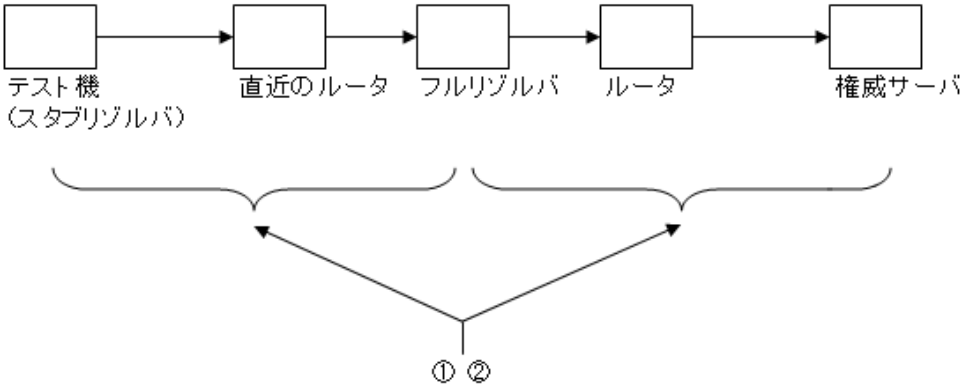
  確認項目 F-147. セキュリティ対応フルリゾルバの IP 層は IPv4 か v6 かに関わら

ず、フラグメントされた UDP パケットを正しく処理できなければならない

- フルリゾルバに誤ったトラストアンカーが設定されているため、フルリゾルバが検証に失敗している。
  シナリオ 8.を参照し、フルリゾルバに適切なトラストアンカーが設定されていることを確認する。

シナリオ 6. フルリゾルバへの問合せに時々失敗する

失敗する状況を確認する。失敗したものと同じ問合せをフルリゾルバに対して何度か行い、常に失敗しているか確認する。



図中の番号はこのシナリオにおいて問題が発生すると考えられる箇所であり、
下記の番号に対応している。この順に参照して問題解決を行うことを想定している。

1. 問い合わせの内容が同じであっても、成功したり失敗したりする

DNSSEC や DNS の問題以前に、テスト機とフルリゾルバの間、あるいはフルリゾルバと権威サーバまでの経路のネットワークの問題である可能性がある。

- シナリオ 5.の 1.～2.を参照してネットワークの問題の切り分けを行い、テスト機からフルリゾルバまでの安定したネットワークの確保を行う。
- フルリゾルバを稼動させているサーバにログインできるのであれば、シナリオ 5.の 3.～4.を参照し、フルリゾルバから権威サーバまでの安定したネットワークの確保を行う。

ここまでの確認項目に問題はないが、このシナリオが解決しない場合は、次に進む。

2. 問い合わせの内容により、必ず成功する(失敗する)ように見える

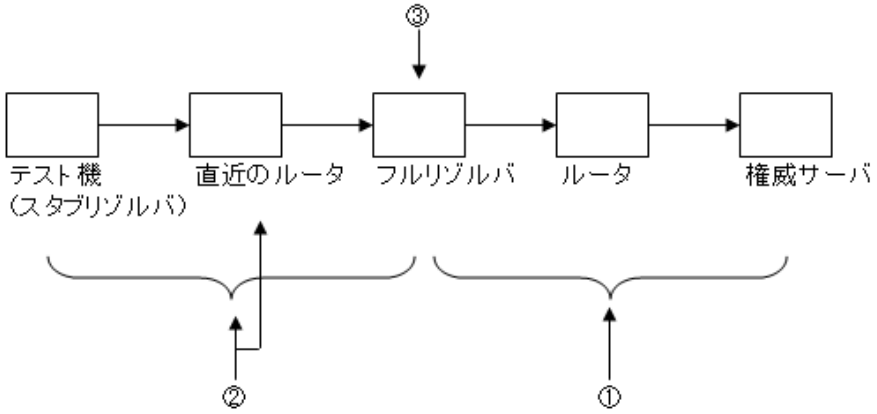
DNSSEC によって問い合わせに対する応答結果のサイズが増えた結果として、問い合わせに対し、あるサイズまでの応答は成功するが、あるサイズを超えるとパケットが

落とされる等の問題が起きている可能性がある。

- フルリゾルバを稼動させているサーバにログインできるのであれば、シナリオ 2.の 2.を参照し、フルリゾルバのサーバから権威サーバに対して直接問い合わせを行い、EDNS0 通信に問題がないかを確認する。
  ここで問題がなければ、権威サーバ側には問題がないことになる。
- 以下を参照し、フルリゾルバが EDNS0 通信が可能かどうかを確認する。
  確認項目 F-85. セキュリティ対応フルリゾルバは EDNS0 による UDP 通信が可能であること

シナリオ 7. フルリゾルバへの問合せが遅い

dig 等によりフルリゾルバへの問い合わせを行うが、応答結果はエラーにならないものの、遅い場合である。
遅くなる原因がフルリゾルバと権威サーバの間にあるのか、テスト機とフルリゾルバの間にあるのかを切り分ける。



図中の番号はこのシナリオにおいて問題が発生すると考えられる箇所であり、下記の番号に対応している。この順に参照して問題解決を行うことを想定している

1. **フルリゾルバから権威サーバに対して直接問い合わせをして、同様に遅いということはないか？**
   フルリゾルバを稼動させているサーバにログインできるのであれば、サーバにログインし、いくつかの権威サーバに対して直接問い合わせを行う。
   - シナリオ 3.を参照し、フルリゾルバと権威サーバとの間の通信に問題がないかを確認する。

   ここまでの確認項目に問題はないが、このシナリオが解決しない場合は、次に進む。

2. テスト機とフルリゾルバとの間のネットワークに問題はないか?

テスト機とフルリゾルバとの間のネットワークに問題がある可能性がある。

- 直近のルータや、テスト機を収容しているネットワークを調査し、ネットワーク障害が起きていないか調べる。
- シナリオ 5.の 1.～2.を参照してネットワークの問題の切り分けを行い、テスト機からフルリゾルバまでのネットワークに問題がないか確認する。

ここまでの確認項目に問題はないが、このシナリオが解決しない場合は、次に進む。

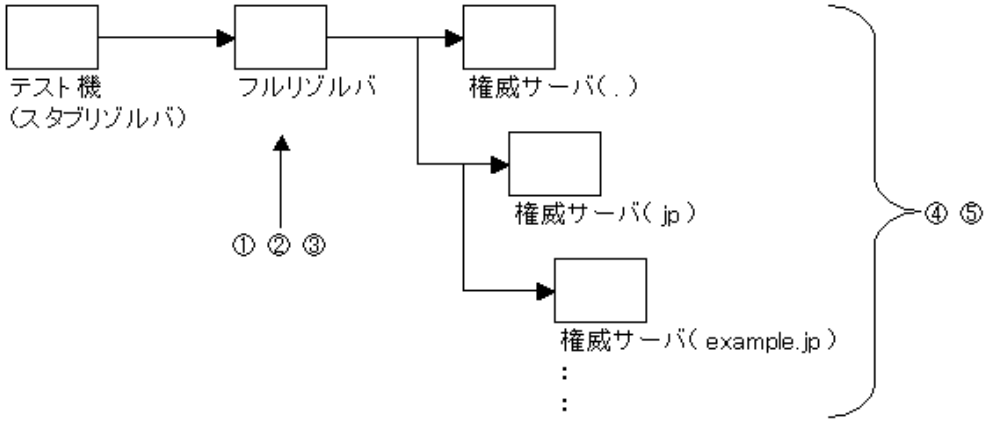
3. フルリゾルバの設定に問題はないか?

フルリゾルバの設定が適切でない可能性がある。フルリゾルバが DNSSEC 対応として正しく設定されているかどうかを確認する。

- シナリオ 5.の 5.を参照して、フルリゾルバの設定を確認する。

シナリオ 8. フルリゾルバが問合せの検証に失敗している

テスト機とフルリゾルバの間、フルリゾルバと権威サーバとの間には問題はないが、DNSSEC 対応の権威サーバが応答した結果について､フルリゾルバが検証に失敗している。そのためスタブリゾルバが行った問い合わせに対する応答がエラーとなっている。あるいは署名済みとならない。



図中の番号はこのシナリオにおいて問題が発生すると考えられる箇所であり、下記の番号に対応している。この順に参照して問題解決を行うことを想定している

1. フルリゾルバの設定において DNSSEC が無効にされてないか？

フルリゾルバの設定において DNSSEC が有効となっていないため、DNSSEC の検証が行われていない可能性がある。

- 以下を参照し、フルリゾルバの設定において DNSSEC が有効となっていることを確認する。

  確認項目 F-130. セキュリティ対応フルリゾルバは元の問い合わせの DO ビットに関わらず、再帰検索においては DO ビットを設定していること

  確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認

2. フルリゾルバに設定されているトラストアンカーに問題はないか？

フルリゾルバに設定されているトラストアンカーが適切ではなく、権威サーバが応答した結果について DNSSEC の検証が失敗している。

- 以下を参照し、フルリゾルバに適切なトラストアンカーが設定されていることを確認する。

  確認項目 F-154. セキュリティ対応フルリゾルバは少なくとも 1 つの信頼できる公開鍵または DS を設定に組み込める機能を持たねばならない

- RFC5011 による自動鍵更新を行う場合は、確認項目 F-201 に従いフルリゾルバの設定ファイルを確認して、適切なトラストアンカーが設定されていることを確認する。

3. フルリゾルバのサーバに設定されている時刻に問題はないか？

フルリゾルバを稼動しているサーバの時刻設定が間違っているため、権威サーバ側の RRSIG レコードの有効期限開始日時、有効期限終了日時が正しくても、フルリゾルバが検証に失敗している。

以下の点を確認する。

- フルリゾルバを稼動しているサーバの時刻設定を、正しい時刻にする。

4. フルリゾルバが権威サーバをたどって目的のゾーンに至るまでの間において、DS レコードと DNSKEY レコードによる信頼の連鎖に問題はないか？

フルリゾルバがルートゾーンから問い合わせを受けたゾーンの権威サーバに至るまでの間のおいて、DS レコードと DNSKEY レコードの設定に問題があり、信頼の連鎖が途切れている。そのため問い合わせが失敗している。

- トラストアンカーで設定されたゾーンから開始して目的のゾーンに至るまでの委任点について、正しく DS が設定されているかを以下の手順を逐次実施して確認する。

  確認項目 A-47. DS レコードのアルゴリズムは対応する DNSKEY レコードのものに

一致するべき

確認項目 A-76. 子ゾーンが署名付きゾーンの場合、委任点に DS レコードが存在すべき

確認項目 A-78. DS は子ゾーンの頂点にあってはならない

確認項目 A-79. DS レコードは子ゾーン頂点の DNSKEY レコードを参照すべき

確認項目 A-81. 子ゾーンの DS の TTL は、委任 NS の TTL と一致すべき

確認項目 A-115. セキュリティ対応権威サーバは委任点の参照を応答する場合、DS とその RRSIG レコードが応答の権威部に含まれることの確認

- トラストアンカーで設定されたゾーンから開始して目的のゾーンに至るまでの各ゾーンについて、正しく DNSKEY が設定されているかを以下の手順を逐次実施して確認する。

確認項目 F-47. DS レコードのアルゴリズムは対応する DNSKEY レコードのものに一致するべき

確認項目 A-58. 署名に使用した鍵がゾーン頂点の DNSKEY レコードに含まれていること

確認項目 A-61. 署名付きゾーンには SEP である DNSKEY RR(KSK 公開鍵情報)が最低 1 つ頂点になければならない

確認項目 A-79. DS レコードは子ゾーン頂点の DNSKEY レコードを参照すべき

5. フルリゾルバが権威サーバをたどって目的のゾーンに至るまで間に、DNSSEC の署名に問題があるゾーンはないか？

フルリゾルバがルートゾーンから問い合わせを受けたゾーンの権威サーバに至るまでの間において、途中の権威サーバでの DNSSEC の署名に問題があり、検証が失敗している。

- トラストアンカーで設定されたゾーンから開始して目的のゾーンに至るまでの各ゾーンについて、DNSSEC の署名に問題がないかどうかを以下の手順を逐次実施して確認する。

確認項目 F-2. DNSSEC 対応フルリゾルバの利用による AD ビットの確認

確認項目 A-27. RRSIG レコードの有効期間終了フィールドが現在時刻より後であること

確認項目 A-28. RRSIG レコードの有効期間開始フィールドが現在時刻より前であること

確認項目 A-58. 署名に使用した鍵がゾーン頂点の DNSKEY レコードに含まれていること

確認項目 A-115. セキュリティ対応権威サーバは委任点の参照を応答する場合、DS とその RRSIG レコードが応答の権威部に含まれることの確認

シナリオ 9. フルリゾルバへの問合せの結果が正しくない

問合せの結果が想定した結果とならなかった場合、何が正しくないのかによって分類する。

1. **権威サーバに直接問い合わせをしてみたが、やはり結果が想定した結果とならない**
   フルリゾルバではなく権威サーバ側に問題がある可能性がある。
   以下の点を確認する。
   - シナリオ 1.～4.を参照し、権威サーバ側を確認する。

2. **権威サーバに直接問い合わせると結果は正しいが、フルリゾルバを介すると結果が正しくない**
   権威サーバが保持しているゾーン情報とフルリゾルバが保持しているゾーン情報に差異が生じている。フルリゾルバが権威サーバの古いゾーン情報をキャッシュしている。
   以下の点を確認する。
   - フルリゾルバを再起動するかキャッシュをフラッシュし、権威サーバが保持する最新のゾーン情報をフルリゾルバにキャッシュさせるようにする。
   - 権威サーバのゾーン情報が更新された場合、権威サーバの SOA レコードのシリアル値が更新されていることを確認する。

IV. 確認項目

1. 権威サーバ側

確認項目 A-27. RRSIG レコードの有効期間終了フィールドが現在時刻より後であること

RRSIG レコードの RDATA の有効期間終了フィールドが現在時刻より後であること。
また、1970 年 1 月 1 日 0 時 0 分 0 秒(UTC)から経過した秒数について、32 ビット符号なし整数あるいは YYYYMMDDHHmmSS の書式であること。

確認項目 A-28. RRSIG レコードの有効期間開始フィールドが現在時刻より前であること

RRSIG レコードの RDATA の有効期間開始フィールドが現在時刻より前であること。
また、1970 年 1 月 1 日 0 時 0 分 0 秒(UTC)から経過した秒数について、32 ビット符号なし整数 あるいは YYYYMMDDHHmmSS の書式であること。

前提事項：

・ DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みであること。

確認方法：

dig コマンドに +dnssec および +norec オプションをつけて、ゾーンの頂点に対する SOA レコードの問い合わせを行う。このとき、以下を指定する。

・ @の後ろには、構築した権威サーバのアドレスを指定する。
・ 構築した権威サーバに格納した、権威あるゾーン名を指定する。（下記の例では、example.jp としている）

得られたレスポンスの ANSWER セクションに SOA レコードに対する RRSIG レコードが含まれていることを確認する(第５カラムが SOA であることを確認する)。また、その RRSIG の有効期間開始時刻（第 10 カラム）が現在の時刻よりも前であり、有効期間終了時刻（第 9 カラム）が現在の時刻よりも後であること(より現実的には検証作業を行う想定の期間よりも後であること)を確認する。

```
$ dig +dnssec +norec @192.0.2.1  example.jp. soa

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.1 example.jp. soa
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36832
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      SOA

;; ANSWER SECTION:
example.jp.             1070    IN      SOA     ns.example.jp. root.example.jp. 2009112800 36
0 90 60480 8640
example.jp.             1070    IN      RRSIG   SOA 7 2 1080 20091230034908 20091130034908 48
272 example.jp. SMv5v4GxorxB3zQKHxQrSEEWCiTH/IIxRxazV8wDxKC080r4q46KNSJ4 pWCrx0v3YcasQnvr042+
sw9Zdin53g==

;; AUTHORITY SECTION:
example.jp.             1032    IN      NS      ns.example.jp.
example.jp.             1032    IN      RRSIG   NS 7 2 1080 20091230034908 20091130034908 482
72 example.jp. q9BGYDXUwDhS5NRUgnnJOPu4naijQdhnK2bXeJQ+w95DKSbghlEnD2ul 4d2XE7CZfPvlWeCqmR5gP
eNGcg+1mg==

;; ADDITIONAL SECTION:
ns.example.jp.          1032    IN      A       192.168.10.1
ns.example.jp.          1070    IN      RRSIG   A 7 3 1080 20091230034908 20091130034908 48272
 example.jp. VzFWniaLaHVi243QzP1CT/yffnMwpOGwHNMQvCy2wLlKuKtBQe4FkU2n 3vjO73MRxlPS2lUgRhdw8GT
jtkpzHw==

;; Query time: 1 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Fri Dec  4 16:02:27 2009
;; MSG SIZE  rcvd: 432
```

なお、dig コマンドで表示される有効期間開始時刻、有効期間終了時刻および、ゾーンファイルや dnssec-signzone の-s あるいは -e オプションで指定する有効期間開始時刻、有効期間終了時刻は JST ではなく UTC なので注意する。

トラブルシューティング：

1. dig の応答がない：

原因 1:
権威サーバが動作していない。権威サーバが動作しているかの確認を行う。

原因 2:
権威サーバまでの経路に問題があり、DNS のパケット(特に DNSSEC の応答パケット)が正しく送られていない。+dnssec を指定しないで dig による問い合わせが成功するかを確認する。

2. RRSIG RR が含まれていない：

原因 1:
権威サーバが正しく DNSSEC 対応として設定されていない。以下の点を確認する。

- dnssec-signzone コマンドでゾーンファイルを署名し、署名したファイルが BIND に読み込まれていること。

原因 2:
権威サーバの設定ファイルにおいて、DNSSEC が無効にされている可能性がある。
以下を参照し、DNSSEC が有効になっていることを確認する。
確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認

3. RRSIG レコードが含まれているが、有効期限期間外である。

原因:
署名が古すぎるか、署名を行った際に指定した有効期限開始日時、有効期限終了日時の指定が正しくない。

以下の 2 点を確認する。

- dnssec-signzone コマンドで再度ゾーンファイルを署名し、署名したファイルが BIND に読み込まれていること。
- dnssec-signzone コマンドで-s オプションあるいは-e オプションを指定する場合は、その値が妥当であることを確認すること。

有効期限開始日時、有効期限終了日時が妥当でない場合には、dnssec-signzone コマンドを実行してゾーンファイルの署名をやり直し、これらの日時が適切になるようにする。

確認項目 A-47. DS レコードのアルゴリズムは対応する DNSKEY レコードのものに一致するべき

DS レコードを検索して得られた結果の RDATA のアルゴリズムフィールドは、参照先の DNSKEY レコードを生成時に選択したアルゴリズムに対応した値となっていること。

確認項目 A-49. DS レコードのダイジェストは対応する DNSKEY レコードの鍵のハッシュであるべき

DS レコードを検索して得られた結果の RDATA のダイジェストフィールドは、参照先のゾーンの DNSKEY KSK 鍵をハッシュした文字列と同じものとなっていること。

確認項目 A-50. DS レコードに対応する DNSKEY レコードはゾーン鍵であるべき

DS レコードの参照する DNSKEY レコードは、DNSSEC ゾーン鍵でなければならない(参照先の DNSKEY レコードの RDATA のフラグフィールドの 7 ビット目に 1 がたっていること)。

前提事項：

- DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みとしていること。
- 署名付きゾーンの DS レコードを、親側の権威サーバに登録済みであること。

ドメインの構成：

- 構築した DNSSEC 対応の権威サーバ(ここでは「構築済みの権威サーバ」と呼びます)がもつ、署名付きの権威あるゾーン名を、sub.example.jp とする。
- 親側のゾーン名を、example.jp とする。
- sub.example.jp は、example.jp の子ゾーンになる。

確認方法：

親側の権威サーバにあらかじめ登録済みである sub.example.jp の DS レコードを確認す

る。親側の権威サーバに対して、dig コマンドに +dnssec および +norec オプションをつけて、sub.example.jp 対する DS レコードの問い合わせを行う。このとき、以下を指定する。

- @の後ろには、親側の権威サーバのアドレスを指定する。
- 構築済みの権威サーバに登録したゾーン名を指定する。
  (下記の例では、sub.example.jp としている)
- レコードタイプは DS を指定する。

```
$ dig +dnssec +norec @192.0.2.1 sub.example.jp. ds

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 sub.example.jp. ds
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1851
;; flags: qr aa; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      DS

;; ANSWER SECTION:
sub.example.jp.         1080    IN      DS      2413 5 1 2475AEAFB5E44FB022082A52468029FB51
370F1A
sub.example.jp.         1080    IN      DS      2413 5 2 97CBDD9120D6E781E265F0F0C4F2672C4B
A5F37E768E81314A4903A9 F49D44D1
sub.example.jp.         1080    IN      RRSIG   DS 5 3 1080 20100105193257 20091206193257 5
2482 example.jp. Mt4QzZZZjS1WgC+NYJS8jDYQqz2pWOKun9Tf7ny4Jy1pbHhEsCkCucQm lU+pUrYbvta9lWf28
TNiH5jGSulPVZiMwse5HzY3igBJ7B4YOPk/YZtH 9flmOQmlswvrqacOnhqLWRzFiTPHl2ffCQQjrunNtmhcdXQWSNh
RsP58 Kdg=

;; Query time: 8 msec
;; SERVER: 192.0.2.1#10053(192.0.2.1)
;; WHEN: Mon Dec  7 05:41:46 2009
;; MSG SIZE  rcvd: 297
```

得られたレスポンスの ANSWER セクションに DS レコードが含まれていることを確認する。

また、DS レコードの鍵タグ(第 5 カラム、この場合 2413)およびアルゴリズム(第 6 カラム、この場合 5)の数値を確認し記録しておく。この例では鍵タグおよびアルゴリズムが同じ DS レコードが二つあるが、これは同じ KSK 公開鍵に対する、異なるダイジェストアルゴリズムを登録してある。

次に、構築済みの権威サーバ(sub.example.jp の権威サーバ)に対して、このゾーンの

DNSKEY レコードを確認する。dig コマンドに +dnssec および +norec をつけて問い合わせを行う。このとき、以下を指定する。

- @の後ろには、sub.example.jp の権威サーバを指定する。
- 署名付きゾーン名を指定する。この例では sub.example.jp となる。
- レコードタイプは DNSKEY を指定する。

```
$ dig +dnssec +norec @192.0.2.2 sub.example.jp. dnskey

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.2 sub.example.jp. dnskey
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2083
;; flags: qr aa; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      DNSKEY

;; ANSWER SECTION:
sub.example.jp.         1080    IN      DNSKEY  257 3 5 AwEAAbmKnHXyXf7hOgv/B5clqWb2VoxADoE
Zxu4PBBxHqe3acQffrAR4 KOewNfw7ESaRHAlG7l2zVLoZ6rtooSDfB/bzuqfzkOdyo8x6GZmgamby 1FQZp4SHGZkj
lOL+tPwg7iLPxvPRw+76AEa89cdgHevZmlJHfn4rGZWj TXVkbTNt
sub.example.jp.         1080    IN      DNSKEY  256 3 5 AwEAAbfYxWvgxlVh1QSdaqOlhBAKLzHtSc+
2vTXB1lhPPsHdlHfHlQeb nxBh2lOlOksnhYlMi5wat7xgQJzpiFn2ZleER/WYAvBeU9edB+uzN6PS LyAjO7CHtOEb
kgtTZ2Zyx/FGS9TcCBSNAiDlrhPeYXmAlnaKl+i+l4Ov yrz9Ttjr
sub.example.jp.         1080    IN      RRSIG   DNSKEY 5 3 1080 20100105200234 200912062002
34 2413 sub.example.jp. jSZQVV8sW17Oq9ik/wspb+K4Q5u5+eWrkEJHZdz6zPcOVUObQAwzwdeW XPUiq5Gl1a
3CqcoMiKV5+JdM1V8zQPKRACoE9ClXOrlBSUdSrkV4SVa9 d/dP6nTDONH9lVxWFgR9rxHtFMeflc/DcRYwslHTbzK+
G8Nnzy4N1Rit MC8=
sub.example.jp.         1080    IN      RRSIG   DNSKEY 5 3 1080 20100105200234 200912062002
34 33018 sub.example.jp. BXFXaZHXJumV8XR5G1J9FLZpxomFXCw61eO/Xr3lGBNEgi98JC4EA6nV wUngw9NxM
8/8nq3Dlk5kNxF+O+PpRskYyjm5C6QpmH4Lb5rdhk2gWHR/ 2N3pnFBEfKrPM2URbPCP3bvuYo5uY88F2g2Qw41FCE6
NdwhOZmVC59Zt 2Fk=

;; Query time: 10 msec
;; SERVER: 192.0.2.2#10053(192.0.2.2)
;; WHEN: Mon Dec  7 06:02:46 2009
;; MSG SIZE  rcvd: 687
```

得られたレスポンスの ANSWER セクションに DNSKEY レコードが含まれていることを確認する。このうち、フラグ(第 5 カラム)が 257 のものが KSK である。この KSK である DNSKEY レコードから DS レコードを生成し、アルゴリズムおよびダイジェストが子ゾーンを検索して得られたものと一致していることを確認すれば、確認項目 47、49、50 を確認できる。

KSK である DNSKEY レコードから DS レコードを生成するために dig によって得られたこの DNSKEY レコードの 1 行だけから成るファイルを作成し、このファイルに対して dnssec-dsfromkey コマンドを実施する。このとき作成するファイルのファイル名は、".key"

で終了している必要がある。

```
$ cat Kexample.key
sub.example.jp.          1080    IN      DNSKEY  257 3 5 AwEAAbmKnHXyXf7hOgv/B5clqWb2VoxADoE
Zxu4PBBxHqe3acQffrAR4 KOewNfw7ESaRHAlG7l2zVLoZ6rtooSDfB/bzuqfzkOdyo8x6GZmgamby 1FQZp4SHGZkj
lOL+tPwg7iLPxvPRw+76AEa89cdgHevZmlJHfn4rGZWj TXVkbTNt
$
$ /usr/local/sbin/dnssec-dsfromkey Kexample.key
sub.example.jp.  IN DS 2413 5 1 2475AEAFB5E44FB022082A52468029FB51370F1A
sub.example.jp.  IN DS 2413 5 2 97CBDD9120D6E781E265F0F0C4F2672C4BA5F37E768E81314A4903A9 F49
D44D1
```

こうして dnssec-dsfromkey で得られた鍵タグ(第 4 カラム、2413)、アルゴリズム（第 5 カラム、5)、ダイジェストアルゴリズム（第 6 カラム、1 および 2)、およびダイジェスト（第 7 カラム以降）が各々親ゾーン example.jp に問い合わせて得られたものと一致していることを確認する。

トラブルシューティング：

1. 構築した権威サーバからのレスポンスに sub.example.jp ゾーンの DNSKEY レコードが含まれていない

   原因:
   権威サーバが正しく DNSSEC 対応として設定されていない。
   以下の点を確認する。
   - 以下を参照し、署名したゾーンファイルが BIND に読み込まれていることを確認する。
     確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認
   - 以下を参照し、DNSSEC が有効になっていることを確認する。
     確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認

2. 構築した権威サーバからのレスポンスに sub.example.jp ゾーンの DNSKEY レコードは含まれているが、鍵文字列が想定したものと異なる

   原因:
   権威サーバが公開している情報と、ゾーンの署名に使用した鍵ファイルが異なっている

可能性がある。以下の点を確認する。

- ゾーンの署名をしたが、そのファイルがまだ BIND に読み込まれていない。あるいは、BIND の named.conf で指定しているゾーンファイルがまちがっている。以下を参照し、署名したゾーンファイルが BIND に読み込まれていることを確認する。

  確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認

3. 親側の権威サーバに DS レコードを問い合わせたが、レスポンスに DS レコードが含まれていない。

   原因:
   親側の権威サーバに、署名付き子ゾーンの DS レコードが登録されていないことが考えられる

   - 確認項目 A-68.のトラブルシューティング 1.を参照し、対応する DS レコードが親側の権威サーバに登録されていることを確認する。

4. 親側の権威サーバが持つ DS レコードの鍵タグ、アルゴリズム、およびダイジェスト本体が、権威サーバから得られた DNSKEY レコードを dnssec-dsfromkey コマンドに入力して得られた値と異なる。

   原因:
   権威あるゾーンの DNSKEY レコードと親側のＤＳレコードが対応していない。

   - 確認項目 A-68.のトラブルシューティング 3.を参照し、対応する DS レコードが親側の権威サーバに登録されていることを確認する。

確認項目 A-58. 署名に使用した鍵がゾーン頂点の DNSKEY レコードに含まれていること

署名したゾーンを持つ権威サーバは、署名に使用した鍵がゾーン頂点の DNSKEY レコード に含まれていること。

前提事項：

・ DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みとしていること。

確認方法：

まず権威サーバを構築時に、権威あるゾーンを署名するときに作成した鍵の公開鍵を確認する。
バックアップに残してある、署名時に dnssec-keygen コマンドで作成した鍵ファイルの内容を確認する。

```
$ cd（鍵ファイルをバックアップしてあるディレクトリ）

$ ls
Kexample.jp.+007+25277.key        ← dnssec-keygen 実行時にZSK 鍵として作成された鍵ファイル
Kexample.jp.+007+25277.private
Kexample.jp.+007+35676.key        ← dnssec-keygen 実行時にKSK 鍵として作成された鍵ファイル
Kexample.jp.+007+35676.private
```

この例では、以下の 2 ファイルが鍵ファイル(公開鍵)になる。

Kexample.jp.+007+25277.key

Kexample.jp.+007+35676.key

このファイルの中身を確認する。

```
$ cat Kexample.jp.+007+25277.key
example.jp.  IN  DNSKEY  256  3  7  AwEAAd5x8zlhOgVtW2zlW1otJmcF5ii2bk/yaUXiDAft/vmkZhWgq8Hh
950etEHE0ho0u/q+twxYLiwtm3S1VY89HmO=

$ cat Kexample.jp.+007+35676.key
example.jp.  IN  DNSKEY  257  3  7  AwEAAcCjW9GtMJKZOtXwXkGmj3tDjSA/6vfwulV4AKH9mZr7yvopmz/w
kil/qaMOl6AeRCpJ4rMMH4AblOhSeLaKaME=
```

ZSK 公開鍵として作成された鍵ファイルでは、DNSKEY レコードの RDATA 部のフラグフィールド(DNKEY の横の数字)が 256 となっている。

KSK 公開鍵として作成された鍵ファイルでは、DNSKEY レコードの RDATA 部のフラグフィールド(DNKEY の横の数字)が 257 となっている。

次に、構築済みの権威サーバに対して、以下の問い合わせを行う。

dig コマンドに +dnssec および +norec オプションをつけて問い合わせを行う。このとき、以下を指定する。

- @の横には、権威サーバを指定する。
- 権威あるゾーン名を指定する。
  (下記の例では、example.jp としている)

```
$ dig +dnssec +norec @192.0.2.1  example.jp DNSKEY

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1  example.jp DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43468
;; flags: qr aa; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      DNSKEY

;; ANSWER SECTION:
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAd5x8zlhOgVtW2zlW1otJmcF5ii2bk
/yaUXiDAft/vmkZhWgq8Hh 95OetEHEOhoOu/q+twxYLiwtm3S1VY89HmO=
example.jp.             1080    IN      DNSKEY  257 3 7 AwEAAcCjW9GtMJKZOtXwXkGmj3tDjSA/6v
fwulV4AKH9mZr7yvopmz/w kil/qaMOl6AeRCpJ4rMMH4AblOhSeLaKaME=
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091218082047 200911180820
47 25277 example.jp. XerM2ZQVAo9ClZMfxgxT/mc5L59BKjcHwB8owjBg9SaWhKc2iwtW56X/ hl3Qevi3yRMVW
x6T6HutMQsCZMOXcg==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091218082047 200911180820
47 35676 example.jp. byYN23vUXUL6W3KjmecM9jULdaFDAvWFrJTeOO/k9jsQSsDNNk1+e6p3 4quhWHukfDuf3
rSOblJaGuLEoMSCmQ==

;; Query time: 1 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Thu Nov 26 19:22:45 2009
;; MSG SIZE  rcvd: 419
```

得られたレスポンスに DNSKEY レコードが含まれており、かつ太字の部分が先ほど確認したファイルの内容と同一であることを確認する。
あわせて、それぞれの DNSKEY レコードのフラグフィールドと鍵文字列の組み合わせも鍵ファイルの内容と同一であることを確認する。

トラブルシューティング：

1. DNSKEY レコードが含まれていない

原因:
権威サーバが正しく DNSSEC 対応として設定されていない。
以下の点を確認する。

- 以下を参照し、署名したゾーンファイルが BIND に読み込まれていることを確認する。
  確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認

- 以下を参照し、DNSSEC が有効になっていることを確認する。
  確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認

2. DNSKEY レコードは含まれているが、鍵文字列が違う

原因:
権威サーバが公開している情報と、ゾーンの署名に使用した鍵ファイルが異なっている可能性がある。
以下の点を確認する。

- ゾーンの署名をしたが、そのファイルがまだ BIND に読み込まれていない。あるいは、BIND の named.conf で指定しているゾーンファイルがまちがっている。以下を参照し、署名したゾーンファイルが BIND に読み込まれていることを確認する。
  確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認

確認項目 A-61. 署名付きゾーンには SEP である DNSKEY RR(KSK 公開鍵情報)が最低 1 つ頂点になければならない

署名したゾーンを持つ権威サーバは、ゾーン頂点の DNSKEY RR のうち、最低 1 つの SEP である DNSKEY RR(KSK 公開鍵情報)を保持していること。

注：
SEP である DNSKEY RR とは、DNSKEY RR のうち、フラグフィールド部が 257 である RR を指す。
また、フラグフィールド部が 257 である DNSKEY RR は、KSK 公開鍵として作成された RR になる。

前提事項：

- DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みとしていること。

確認方法：

構築済みの権威サーバに対して、以下の問い合わせを行う。

dig コマンドに +dnssec および +norec オプションをつけて問い合わせを行う。このとき、以下を指定する。
- @の横には、権威サーバを指定する。
- 権威あるゾーン名を指定する。
  (下記の例では、example.jp としている)

```
$ dig +dnssec +norec @192.0.2.1  example.jp DNSKEY

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1  example.jp DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43468
;; flags: qr aa; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      DNSKEY

;; ANSWER SECTION:
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAd5x8zlhOgVtW2zlW1otJmcF5ii2bk/
yaUXiDAft/vmkZhWgq8Hh 95OetEHEOhoOu/q+twxYLiwtm3S1VY89HmO=
example.jp.             1080    IN      DNSKEY  257 3 7 AwEAAcCjW9GtMJKZOtXwXkGmj3tDjSA/6vf
wulV4AKH9mZr7yvopmz/w kil/qaMOl6AeRCpJ4rMMH4AblOhSeLaKaME=
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091218082047 200911180820
47 25277 example.jp. XerM2ZQVAo9ClZMfxgxT/mc5L59BKjcHwB8owjBg9SaWhKc2iwtW56X/ hl3Qevi3yRMVW
x6T6HutMQsCZMOXcg==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091218082047 200911180820
47 35676 example.jp. byYN23vUXUL6W3KjmecM9jULdaFDAvWFrJTeOO/k9jsQSsDNNk1+e6p3 4quhWHukfDuf3
rSOblJaGuLEoMSCmQ==

;; Query time: 1 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Thu Nov 26 19:22:45 2009
;; MSG SIZE  rcvd: 419
```

得られたレスポンスに DNSKEY RR が含まれていることを確認する。
また、そのうち最低 1 つが DNSKEY RR のフラグフィールドが 257 となっていることを確認する(フラグフィールドが 257 であれば、それは SEP である DNSKEY RR であることを意味する)。

トラブルシューティング：

1. DNSKEY RR が含まれていない

   原因:
   権威サーバが正しく DNSSEC 対応として設定されていない。
   以下の点を確認する。

   - 以下を参照し、署名したゾーンファイルが BIND に読み込まれていることを確認する。
     確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認

   - 以下を参照し、DNSSEC が有効になっていることを確認する。

確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認

2. DNSKEY RR は含まれているが、SEP であるレコードが含まれていない
(フラグフィールドが 256 である DNSKEY RR しか含まれていない、など)

原因:
以下の 2 点が考えられる。

- ゾーンの署名時に、dnssec-keygen コマンドで KSK 公開鍵を作成していない
- あるいは、KSK 公開鍵を作成したが作成した鍵ファイルの内容をゾーンファイルに追記しないまま、ゾーンの署名を行っている

以下の点を確認する。

- dnssec-keygen コマンドにて、 KSK 公開鍵を作成してあること。
- KSK 公開鍵を作成しており、かつ署名後のゾーンファイルにも反映されていること(反映されていなければ、作成した鍵ファイルの内容をゾーンファイルに追記しないまま、ゾーンの書名を作成していたことになる)
- 署名したファイルが BIND に読み込まれていること。
以下を参照し、署名したゾーンファイルが BIND に読み込まれていることを確認する。
確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認

確認項目 A-68. 署名付きゾーン頂点の DNSKEY と親側の委任点にある DS が示すアルゴリズムの確認

署名付きゾーン頂点の DNSKEY 自身は親側の委任点にある DS が示すアルゴリズムによって署名されていなければならない。

確認項目 A-76. 子ゾーンが署名付きゾーンの場合、委任点に DS レコードが存在すべき

子ゾーンが署名付きゾーンの場合、親側権威サーバの委任点には、子ゾーンの DS レコードが存在すべき。

前提事項：

- DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みとしていること。
- 署名付きゾーンの DS レコードを、親側の権威サーバに登録済みであること。

ドメインの構成：

- 構築した DNSSEC 対応の権威サーバ(ここでは「構築済みの権威サーバ」と呼ぶ)がもつ、署名付きの権威あるゾーン名を、sub.example.jp とする。
- 親側のゾーン名を、example.jp とする。
- sub.example.jp は、example.jp の子ゾーンになる。

確認方法：

まず親側の権威サーバに、あらかじめ登録済みである sub.example.jp の DS レコードを確認する。
親側の権威サーバに対して、以下の問い合わせを行う。

dig コマンドに +dnssec および +norec オプションをつけて問い合わせを行う。このとき、以下を指定する。

- @の横には、親側の権威サーバを指定する。
- 署名付きゾーン名を指定する。この例では sub.example.jp となる。
- レコードタイプは DS を指定する。

```
$ dig +dnssec +norec @192.0.2.1  sub.example.jp DS

<<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 sub.example.jp DS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27176
;; flags: qr aa; QUERY: 1, ANSWER: 3, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      DS

;; ANSWER SECTION:
sub.example.jp.         1080    IN      DS      23454 7 2 B49C5A8AE492A44BBDA45908E114FB5C7
9AEF74F4BE7E5FEDB5312FF C120ACF3
sub.example.jp.         1080    IN      DS      23454 7 1 2D35FC7B0197A150A8D8958964C983259
EC271E1
sub.example.jp.         1080    IN      RRSIG   DS 7 3 1080 20091227053717 20091127053717 4
8272 example.jp. q909fN59Z+Xplt7vfcRXOOm4aVy4mjBju1G55vRqco5Hp1BAmqZXGwYR TwbFDl4/BKCL6EiGw
2+egWzEYc/olg==

;; AUTHORITY SECTION:
example.jp.             1080    IN      NS      ns.example.jp.
example.jp.             1080    IN      RRSIG   NS 7 2 1080 20091227053717 20091127053717 4
8272 example.jp. MTdauvE1dMaHZE/litFXRSUZFZU+v78oHEhLKcscecjBKCwei9qsNB6X +By2eDolcwkYPH9PF
pzYaRGTqUyeDA==

;; ADDITIONAL SECTION:
ns.example.jp.          1080    IN      A       192.0.2.1
ns.example.jp.          1080    IN      RRSIG   A 7 3 1080 20091227053717 20091127053717 482
72 example.jp. aJcHslgP6nd78Ym5MuiMNRYttcb/6yloiCtBhT/W+kMJklozhQ1MG6NN sAB9xbrPjnjOFbYkr/Q
A6z49GDTtKw==

;; Query time: 2 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Fri Nov 27 16:52:51 2009
;; MSG SIZE  rcvd: 479
```

得られたレスポンスの ANSWER SECTION に DS レコードが含まれていることを確認する。
また、DS レコードの RDATA 部のアルゴリズムフィールドの数値(この例では、23454 の横の 7)を確認しておく。

次に、構築済みの権威サーバ(sub.example.jp の権威サーバ)に対して、このゾーンの DNSKEY レコードを確認する。
以下の問い合わせを行う。

dig コマンドに +dnssec および +norec オプションをつけて問い合わせを行う。このとき、以下を指定する。

・ @の横には、sub.example.jp の権威サーバを指定する。

・ 署名付きゾーン名を指定する。この例では sub.example.jp となる。
・ レコードタイプは DNSKEY を指定する。

```
$ dig +dnssec +norec @192.0.20.101 sub.example.jp DNSKEY

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.20.101 sub.example.jp DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24062
;; flags: qr aa; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      DNSKEY

;; ANSWER SECTION:
sub.example.jp.         1080    IN      DNSKEY  256 3 7 AwEAAdwtWnQ53O4EMg6wKITO3qDhrsDj7/S
lWgvwtpNOmrP9JCaAExrm mSlT+zOACCBeZCkVa932CF1l6SrM+a3FAns=
sub.example.jp.         1080    IN      DNSKEY  257 3 7 AwEAAcEHNbl/iHv9qRd14VMx9uekrzDBc9z
ieWPqfo1oObx3lhwzLqzC 648pOGyRaElgH1RNWjMDiyk2SJ7ah186KE8=
sub.example.jp.         1080    IN      RRSIG   DNSKEY 7 3 1080 20091227052347 200911270523
47 23454 sub.example.jp. YcoaJiw0xJDGT3LosxyZtiNvcSrdpo/2inBol0RIbo5HNAC2WITVxd4g Ex0LM8QIb
4x1azzC3RuuVrjrBGvBGw==
sub.example.jp.         1080    IN      RRSIG   DNSKEY 7 3 1080 20091227052347 200911270523
47 27878 sub.example.jp. efYaLBQmDmWL6B6amgJE1/Z52Evldvd//N29NXV7jOd4c3UEJgMq8H6a qNm1HbPqv
ER3ryKg7GnEPTU69XBjYw==

;; Query time: 3 msec
;; SERVER: 192.0.20.101#53(192.0.20.101)
;; WHEN: Fri Nov 27 17:17:13 2009
;; MSG SIZE  rcvd: 431
```

得られたレスポンスの ANSWER SECTION に DNSKEY レコードが含まれていることを確認する。
複数得られた DNSKEY レコードのうち、RDATA 部のフラグフィールド(DNSKEY という文字列のすぐ横)が 257 であるレコードの、アルゴリズムフィールド(フラグフィールドの 2 つ右横、この例では 7 となっているところ)の数値を確認する。

この数値が、先ほど確認した親側の権威サーバが保持している DS レコードのアルゴリズムフィールドの数値と同じになっていることを確認する。

注意点：
得られた DNSKEY レコードのうち、フラグフィールドが 257 以外のレコードは確認に用いてはならない。

トラブルシューティング：

1. 親側の権威サーバに DS レコードを問い合わせたが、レスポンスに DS レコードが含まれていない

   原因:
   親側の権威サーバに、署名付きゾーンの DS レコードが登録されていないことが考えられる。
   以下の点を確認する。
   - 署名付きゾーンの DS レコードを、親側の権威サーバに登録し忘れていないか。
   - 署名付きゾーンの DS レコードを親側の権威サーバに登録する際に、署名付きゾーンのゾーン名が間違っていないか。
     この例では、DS レコードのゾーン名は sub.example.jp となる。これとは異なったゾーン名で登録した場合、上記の dig コマンドでは問い合わせ結果に DS レコードが含まれない。

2. 構築済みの権威あるサーバ(sub.example.jp の権威サーバ)に DNSKEY レコードを問い合わせたが、レスポンスに DNSKEY レコードが含まれていない。あるいは、レスポンスに含まれている DNSKEY レコードのうち、フラグフィールドが 257 であるレコードが含まれていない(フラグフィールドが 256 である DNSKEY レコードしか含まれていない、など)

   原因:
   以下の 2 点が考えらる。
   - ゾーンの署名時に、dnssec-keygen コマンドで KSK 公開鍵を作成していない
   - あるいは、KSK 公開鍵を作成したが作成した鍵ファイルの内容をゾーンファイルに追記しないまま、ゾーンの署名を行っている

   以下の点を確認する。
   - dnssec-keygen コマンドにて KSK 公開鍵を作成してあること。
   - KSK 公開鍵を作成しており、かつ署名後のゾーンファイルにも反映されていること(反映されていなければ、作成した鍵ファイルの内容をゾーンファイルに追記しないまま、ゾーンの書名を作成していたことになる)
   - 署名したファイルが BIND に読み込まれていること。
     以下を参照し、署名したゾーンファイルが BIND に読み込まれていることを確認する。
     確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認

3. 親側の権威サーバが持つ DS レコードのアルゴリズムフィールドと、構築済みの権威あるサーバが持つ DNSKEY レコードのアルゴリズムフィールドの数値が違っている

原因:
権威あるゾーンの DNSKEY レコードと親側のＤＳレコードが対応していない。
以下のような手順ミスが起きていることが考えられる。

- 以前に権威あるゾーン(この例では sub.example.jp ゾーン)の KSK 公開鍵を作成し、DNSKEY レコードを構築済みの権威サーバに登録し、対応する DS レコードも親側の権威サーバにも登録した。
  その後、違うアルゴリズムを用いて権威あるゾーンの KSK 公開鍵を作成し、DNSKEY レコードを構築済みの権威サーバに登録したが、対応する DS レコードを親側の権威サーバに登録していない。
  あるいは、対応する DS レコードは親側の権威サーバに登録したが、KSK 公開鍵の DNSKEY レコードを構築済みの権威サーバに登録していない。

以下の点を確認する。

- あらたに KSK 公開鍵を作成しているのであれば、対応する DS レコードを親側の権威サーバに登録していること。あるいは、作成した KSK 公開鍵を構築済みの権威サーバに登録していること。
- 作成した KSK 公開鍵が構築済みの権威サーバに正しく登録されているかどうかは、上記トラブルシューティング 2.の項目を確認する。

確認項目 A-78. DS は子ゾーンの頂点にあってはならない

子ゾーンが署名付きゾーンの場合、子ゾーンの DS レコードは、子ゾーンの頂点に存在してはならない。

前提事項：

・ DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みとしていること。

ドメインの構成：

・ 構築した DNSSEC 対応の権威サーバ(ここでは「構築済みの権威サーバ」と呼ぶ)がもつ、署名付きの権威あるゾーン名を、sub.example.jp とする。

確認方法：

構築済みの権威サーバ(sub.example.jp の権威サーバ)に対して、このゾーンの DS レコードを確認する。
以下の問い合わせを行う。

dig コマンドに +dnssec および +norec オプションをつけて問い合わせを行う。このとき、以下を指定する。
・ @の横には、sub.example.jp の権威サーバを指定する。
・ 署名付きゾーン名を指定する。この例では sub.example.jp となる。
・ レコードタイプは DS を指定する。

```
$ dig +dnssec +norec @192.0.2.101 sub.example.jp DS

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.101 sub.example.jp DS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57332
;; flags: qr aa; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      DS

;; AUTHORITY SECTION:
sub.example.jp.         1080    IN      SOA     ns.sub.example.jp. root.example.jp. 2009112
703 360 90 60480 8640
sub.example.jp.         1080    IN      RRSIG   SOA 7 3 1080 20091227052347 20091127052347
27878 sub.example.jp. XTjC1QZLOpCqQNZpY/5neFO6r+UElYJbnwSXGe369lNqw+cdlt+Of/3a oNq+OMq/JJrD
OPg7hGOQWy8LO/LPXQ==
MEDOGC5QG5DRTEF55B22726N6PDN1PMQ.sub.example.jp. 8640 IN NSEC3 1 0 100 AAAA MSRKP3LUSRFGP9K
RLEO3E2O1L9UV6VLL NS SOA RRSIG DNSKEY NSEC3PARAM
MEDOGC5QG5DRTEF55B22726N6PDN1PMQ.sub.example.jp. 8640 IN RRSIG NSEC3 7 4 8640 2009122705234
7 20091127052347 27878 sub.example.jp. jB9Sd7jl7EJgsNbUmPer0VCv3KpriBwXXw5yl6oxlpK5pWK/0D4P
YUDK c8Z8pNh4aoRMZXU/D3HYuUu7juUbJg==

;; Query time: 2 msec
;; SERVER: 192.0.2.101#53(192.0.2.101)
;; WHEN: Fri Nov 27 19:46:34 2009
;; MSG SIZE  rcvd: 390
```

得られたレスポンスには、sub.example.jp に対する DS レコードが含まれていないことを確認する。
また権威サーバが DNSSEC 対応として正しく設定されている場合、問い合わせられたレコードが存在しない場合は、代わりにレスポンスに NSEC3 レコードが含まれるようになるので、そのことを確認する。
なお、DNSSEC の不在証明を NSEC3 レコード形式ではなく NSEC レコード形式で行っている場合には、レスポンスに NSEC レコードが含まれるようになるので、そのことを確認する。

トラブルシューティング：

1. 得られたレスポンスに sub.example.jp に対する DS レコードが含まれている

   原因:
   構築済みの権威サーバに、権威あるゾーンの DS レコードが登録されてる。DS レコードは親側の権威サーバに登録するものであり、権威あるゾーンには登録してはならない。
   以下を行う。

- 構築済みの権威サーバのゾーンファイルから DS レコードを削除し、dnssec-signzone コマンドをやり直して BIND に再読み込みさせる。
- DS レコードは正しくは親側の権威サーバに登録するものである。親側の権威サーバへの登録がまだなら、登録を行う。

確認項目 A-79. DS レコードは子ゾーン頂点の DNSKEY レコードを参照すべき

親側の権威サーバが保持している子ゾーンの DS レコードは、署名付きの子ゾーン頂点にある DNSKEY レコードと対応が取れていること。

確認項目 A-80. 署名付きの子ゾーン頂点にある DNSKEY レコードは、対応する DS レコードと同じ秘密鍵で署名されるべき

署名付きの子ゾーン頂点にある DNSKEY レコードは、親側の権威サーバに登録されている DS レコードと同じ秘密鍵で署名されていること。

前提事項：

- DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みとしていること。
- 署名付きゾーンの DS レコードを、親側の権威サーバに登録済みであること。

ドメインの構成：

- 構築した DNSSEC 対応の権威サーバ(ここでは「構築済みの権威サーバ」と呼ぶ)がもつ、署名付きの権威あるゾーン名を、sub.example.jp とする。
- 親側のゾーン名を、example.jp とする。
- sub.example.jp は、example.jp の子ゾーンになる。

確認方法：

まず、権威サーバにおいて、権威あるゾーンを署名するときに作成した鍵の公開鍵のうち KSK 公開鍵として作成された鍵を確認する。

具体的な手順は下記のとおり。

1.
バックアップに残してある、署名時に dnssec-keygen コマンドで作成した鍵ファイルの内容を確認する。

```
$ cd（鍵ファイルをバックアップしてあるディレクトリ）

$ ls
Ksub.example.jp.+007+23454.key
        ↑ dnssec-keygen 実行時にKSK 鍵として作成された鍵ファイル
Ksub.example.jp.+007+23454.private
Ksub.example.jp.+007+27878.key
        ↑ dnssec-keygen 実行時にZSK 鍵として作成された鍵ファイル
Ksub.example.jp.+007+27878.private
```

この例では、以下の 2 ファイルが鍵ファイル(公開鍵)になる。

Ksub.example.jp.+007+23454.key

Ksub.example.jp.+007+27878.key

2.

このファイルの中身を確認する。

```
$ cat Ksub.example.jp.+007+23454.key
sub.example.jp.  IN DNSKEY 257 3 7 AwEAAcEHNbI/iHv9qRd14VMx9uekrzDBc9zieWPqfo1oObx3IhwzLqzC
 648pOGyRaEIgH1RNWjMDiyk2SJ7ah186KE8=

$ cat Ksub.example.jp.+007+27878.key
sub.example.jp.  IN DNSKEY 256 3 7 AwEAAdwtWnQ53O4EMg6wKIT03qDhrsDj7/SIWgvwtpNOmrP9JCaAExrm
mSIT+zOACCBeZCkVa932CF1I6SrM+a3Fans=
```

ZSK 公開鍵として作成された鍵ファイルでは、DNSKEY レコードの RDATA 部のフラグフィールド(DNSKEY の横の数字)が 256 となっている。
KSK 公開鍵として作成された鍵ファイルでは、DNSKEY レコードの RDATA 部のフラグフィールド(DNSKEY の横の数字)が 257 となっている。

この例では、Ksub.example.jp.+007+23454.key が KSK 公開鍵となる。
鍵タグは 23454 となる(ファイル名の +007+ と .key の間の数字)。
また、Ksub.example.jp.+007+27878.key が ZSK 公開鍵となる。
鍵タグは 23878 となる。

次に、構築済みの権威サーバに対してこの DNSKEY レコードが登録されているかを確認する。以下の問い合わせを行う。

dig コマンドに +dnssec および +norec オプションをつけて問い合わせを行う。このとき、以下を指定する。

- @の横には、権威サーバを指定する。
- 権威あるゾーン名を指定する。

(下記の例では、sub.example.jp としている)

・ レコードタイプは DNSKEY を指定する。

```
$ dig +dnssec +norec @192.0.2.101 sub.example.jp DNSKEY

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.101 sub.example.jp DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40212
;; flags: qr aa; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      DNSKEY

;; ANSWER SECTION:
sub.example.jp.         1080    IN      DNSKEY  257 3 7 AwEAAcEHNbI/iHv9qRd14VMx9uekrzDBc9
zieWPqfo1oObx3IhwzLqzC 648pOGyRaEIgH1RNWjMDiyk2SJ7ah186KE8=
sub.example.jp.         1080    IN      DNSKEY  256 3 7 AwEAAdwtWnQ53O4EMg6wKITO3qDhrsDj7/
SIWgvwtpNOmrP9JCaAExrm mSIT+zOACCBeZCkVa932CF1l6SrM+a3FAns=
sub.example.jp.         1080    IN      RRSIG   DNSKEY 7 3 1080 20091227052347 200911270523
47 23454 sub.example.jp. YcoaJiw0xJDGT3LosxyZtiNvcSrdpo/2inBolORlbo5HNAC2WITVxd4g ExOLM8QIb
4x1azzC3RuuVrjrBGvBGw==
sub.example.jp.         1080    IN      RRSIG   DNSKEY 7 3 1080 20091227052347 200911270523
47 27878 sub.example.jp. efYaLBQmDmWL6B6amgJE1/Z52Evldvd//N29NXV7jOd4c3UEJgMq8H6a qNm1HbPqv
ER3ryKg7GnEPTU69XBjYw==

;; Query time: 1 msec
;; SERVER: 192.0.2.101#53(192.0.2.101)
;; WHEN: Wed Dec  2 21:02:36 2009
;; MSG SIZE  rcvd: 431
```

得られたレスポンスに KSK 公開鍵である DNSKEY レコードが含まれており、かつ太字の部分(DNSKEY レコードのフラグフィールドの値、鍵文字列)が先ほど確認した KSK 公開鍵ファイルの内容と同一であることを確認する。このとき、以下の点を確認する。

次に、親側の権威サーバに、あらかじめ登録済みである sub.example.jp の DS レコードを確認する。
親側の権威サーバに対して、以下の問い合わせを行う。

dig コマンドに +dnssec および +norec オプションをつけて問い合わせを行う。このとき、以下を指定する。

・ @の横には、親側の権威サーバを指定する。
・ 署名付きゾーン名を指定する。この例では sub.example.jp となる。
・ レコードタイプは DS を指定する。

```
$ dig +dnssec +norec @192.0.2.1  sub.example.jp DS

<<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.1 sub.example.jp DS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27176
;; flags: qr aa; QUERY: 1, ANSWER: 3, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      DS

;; ANSWER SECTION:
sub.example.jp.         1080    IN      DS      23454 7 2 B49C5A8AE492A44BBDA45908E114FB5C7
9AEF74F4BE7E5FEDB5312FF C120ACF3
sub.example.jp.         1080    IN      DS      23454 7 1 2D35FC7B0197A150A8D8958964C983259
EC271E1
sub.example.jp.         1080    IN      RRSIG   DS 7 3 1080 20091227053717 20091127053717 4
8272 example.jp. q909fN59Z+Xplt7vfcRXOOm4aVy4mjBju1G55vRqco5Hp1BAmqZXGwYR TwbFDl4/BKCL6EiGw
2+egWzEYc/olg==

;; AUTHORITY SECTION:
example.jp.             1080    IN      NS      ns.example.jp.
example.jp.             1080    IN      RRSIG   NS 7 2 1080 20091227053717 20091127053717 4
8272 example.jp. MTdauvE1dMaHZE/litFXRSUZFZU+v78oHEhLKcscecjBKCwei9qsNB6X +By2eDolcwkYPH9PF
pzYaRGTqUyeDA==

;; ADDITIONAL SECTION:
ns.example.jp.          1080    IN      A       192.0.2.1
ns.example.jp.          1080    IN      RRSIG   A 7 3 1080 20091227053717 20091127053717 482
72 example.jp. aJcHslgP6nd78Ym5MuiMNRYttcb/6yloiCtBhT/W+kMJklozhQ1MG6NN sAB9xbrPjnjOFbYkr/Q
A6z49GDTtKw==

;; Query time: 2 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Fri Nov 27 16:52:51 2009
;; MSG SIZE  rcvd: 479
```

得られたレスポンスの ANSER SECTION に DS レコードが含まれていることを確認する。また、DS レコードの RDATA 部の鍵タグフィールドの数値(DNSKEY という文字列のすぐ横、この例では、23454)を確認する。

この鍵タグの数字が、先ほど権威サーバ側で確認した KSK 公開鍵のファイル名に含まれている鍵タグの数字と等しいことを確認する。
鍵タグの数字が等しければ、権威サーバ側(署名付きの子ゾーン)の DNSKEY レコードと、親側の権威サーバの DS レコードの対応が取れていることになる。

注意点：
得られた DNSKEY レコードのうちフラグフィールドが 257 以外のレコードは、確認に用いてはならない。

トラブルシューティング：

1. 構築した権威サーバからのレスポンスに sub.example.jp ゾーンの DNSKEY レコードが含まれていない

   原因:
   権威サーバが正しく DNSSEC 対応として設定されていない。
   以下の点を確認する。
   - 以下を参照し、署名したゾーンファイルが BIND に読み込まれていることを確認する。
     確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認

   - 以下を参照し、DNSSEC が有効になっていることを確認する。
     確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認

2. 構築した権威サーバからのレスポンスに sub.example.jp ゾーンの DNSKEY レコードは含まれているが、鍵文字列が違う

   原因:
   権威サーバが公開している情報と、ゾーンの署名に使用した鍵ファイルが異なっている可能性がある。
   以下の点を確認する。
   - ゾーンの署名をしたが、そのファイルがまだ BIND に読み込まれていない。あるいは、BIND の named.conf で指定しているゾーンファイルがまちがっている。
     以下を参照し、署名したゾーンファイルが BIND に読み込まれていることを確認する。
     確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認

3. 親側の権威サーバに DS レコードを問い合わせたが、レスポンスに DS レコードが含まれていない。

   原因:
   親側の権威サーバに、署名付き子ゾーンの DS レコードが登録されていないことが考え

られる。

- 確認項目 A-68.のトラブルシューティング 1.を参照し、対応する DS レコードが親側の権威サーバに登録されていることを確認する。

4. 親側の権威サーバが持つ DS レコードの鍵タグフィールドが、権威サーバ側で確認した KSK 公開鍵のファイル名に含まれている鍵タグの数字と一致しない。

原因:
権威あるゾーンの DNSKEY レコードと親側のＤＳレコードが対応していない。

- 確認項目 A-68.のトラブルシューティング 3.を参照し、対応する DS レコードが親側の権威サーバに登録されていることを確認する。

確認項目 A-81. 子ゾーンの DS の TTL は、委任 NS の TTL と一致すべき

親側の権威サーバの委任点に保持している子ゾーンの DS レコードの TTL は、同様に委任点に保持している子ゾーンの NS レコードの TTL と一致しているべき。

前提事項：

- DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みとしていること。
- 署名付きゾーンの DS レコードを、親側の権威サーバに登録済みであること。

ドメインの構成：

- 構築した DNSSEC 対応の権威サーバ(ここでは「構築済みの権威サーバ」と呼ぶ)がもつ、署名付きの権威あるゾーン名を、sub.example.jp とする。
- 親側のゾーン名を、example.jp とする。
- sub.example.jp は、example.jp の子ゾーンになる。

確認方法：

親側の権威サーバに、あらかじめ登録済みである sub.example.jp の DS レコードと NS レコードを確認する。
親側の権威サーバに対して、以下の問い合わせを行う。

dig コマンドに +dnssec オプションをつけて問い合わせを行う。このとき、以下を指定する。

- +norec オプションも指定する。
  - 親側の権威サーバに子ゾーンの NS レコードを問い合わせると、子ゾーンの権威サーバに再帰問い合わせを行ってしまう。ここでは親側の権威サーバが保持している委任点の NS レコードを確認したいので、再帰問い合わせを行わないようにするために指定している。
- @の横には、親側の権威サーバを指定する。
- 子ゾーン名を指定する。この例では sub.example.jp となる。
- レコードタイプは NS を指定する。

```
$ dig +dnssec +norec @192.0.2.1 sub.example.jp NS

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 sub.example.jp NS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35667
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      NS

;; AUTHORITY SECTION:
sub.example.jp.         1080    IN      NS      ns.sub.example.jp.
sub.example.jp.         1080    IN      DS      23454 7 1 2D35FC7B0197A150A8D8958964C983259
EC271E1
sub.example.jp.         1080    IN      DS      23454 7 2 B49C5A8AE492A44BBDA45908E114FB5C7
9AEF74F4BE7E5FEDB5312FF C120ACF3
sub.example.jp.         1080    IN      RRSIG   DS 7 3 1080 20091227053717 20091127053717 4
8272 example.jp. q909fN59Z+Xplt7vfcRXOOm4aVy4mjBju1G55vRqco5Hp1BAmqZXGwYR TwbFDl4/BKCL6EiGw
2+egWzEYc/olg==

;; ADDITIONAL SECTION:
ns.sub.example.jp.      1080    IN      A       192.0.2.101

;; Query time: 0 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Fri Nov 27 20:26:45 2009
;; MSG SIZE  rcvd: 266
```

得られたレスポンスの AUTHORITY SECTION に DS レコードと NS レコードが含まれていることを確認する。
得られた DS レコードと NS レコードの TTL 値が同一であることを確認する。

トラブルシューティング：

1. 親側の権威サーバに DS レコードを問い合わせたが、レスポンスに DS レコードが含まれていない

   原因:
   親側の権威サーバに、子ゾーンの DS レコードが登録されていないことが考えられる。
   確認項目 A-076. **子ゾーンが署名付きゾーンの場合、委任点に DS レコードが存在すべき** を参照し、DS レコードが登録されているかどうかを確認する。

2. 親側の権威サーバに、子ゾーンの NS レコードが含まれていない

   原因:
   親側の権威サーバのゾーンファイルに、子ゾーンの NS レコードが設定されていないこ

とが考えられる。

以下の点を確認する。

- 親側の権威サーバのゾーンファイルに、子ゾーン(この場合は sub.example.jp)の NS レコードが設定されていること。

  例：
  親側の権威サーバのゾーンファイルには、以下の一行が記述されていること。

  sub.example.jp. 1080 IN NS ns.sub.example.jp.

3. 得られた DS レコードと NS レコードの TTL 値が異なっている

原因:
親側の権威サーバのゾーンファイルにおいて、子ゾーンの DS レコードと NS レコードの TTL 値が正しく設定されていない。

以下の点を確認する。

- 親側の権威サーバのゾーンファイルにおいて、子ゾーンの DS レコードと NS レコードの TTL 値が同値となるように設定されていること。

  例：
  ここでは親側の権威サーバのゾーンファイルを編集する際に、子ゾーンの DS レコードと NS レコードの TTL 値を同値となるように設定する手順の一例を説明する。

  1.
  親側の権威サーバの(署名前の)ゾーンファイルを作成する前に、BIND のゾーンファイルの省略記法を利用して、以下のように各レコードには TTL 値を記述しないように作成する。

```
$TTL 1080
example.jp.             IN      SOA ns.example.jp. root.example.jp. (
                      2009112705 ; serial
                      360        ; refresh 1hr
                      90         ; retry 15min
                      60480      ; expire 1w
                      8640       ; min 24hr
)
example.jp.               IN      NS      ns.example.jp.

～略～

sub.example.jp.           IN      NS      ns.sub.example.jp.
ns.sub.example.jp.        IN      A       192.0.2.101
```

2.

次に、親側の権威サーバのゾーンファイルに、子ゾーンの DS レコードを追記する。

子ゾーン側でゾーンファイルを dnssec-signzone コマンドで署名した際に、dsset-(ゾーン名)というファイルが作成されているはずである(この例では、dsset-sub.example.jp というファイル名になる)。

このファイルには、親側のゾーンに登録する DS レコードが記述されており、下記のように TTL 値が記述されていない。

```
sub.example.jp.           IN DS 23454 7 1 2D35FC7B0197A150A8D8958964C983259EC
271E1
sub.example.jp.           IN DS 23454 7 2 B49C5A8AE492A44BBDA45908E114FB5C79A
EF74F4BE7E5FEDB5312FF C120ACF3
```

子ゾーン側からこのファイルの内容を受け取れる場合は、これを利用する。

このファイルの内容をこのまま親側の権威サーバのゾーンファイルに追記する。

3.

この状態で親側の権威サーバのゾーンファイルを dnssec-signzone コマンドで署名すると、以下のように子ゾーンの DS レコードと NS レコードの TTL 値をそろえることができる。

```
; File written on Fri Nov 27 15:37:17 2009
; dnssec_signzone version 9.6.1-P1
example.jp.       1080    IN SOA  ns.example.jp. root.example.jp. (
                                  2009112705 ; serial
                                  360        ; refresh (6 minutes)
                                  90         ; retry (1 minute 30 seconds)
                                  60480      ; expire (16 hours 48 minutes)
                                  8640       ; minimum (2 hours 24 minutes)
                                  )
～略～
ns.sub.example.jp.  1080    IN A    192.0.2.101
sub.example.jp.     1080    IN NS   ns.sub.example.jp.
                    1080    DS      23454 7 1 (
                                    2D35FC7B0197A150A8D8958964C983259EC2
                                    71E1 )
                    1080    DS      23454 7 2 (
                                    B49C5A8AE492A44BBDA45908E114FB5C79AE
                                    F74F4BE7E5FEDB5312FFC120ACF3 )
                    1080    RRSIG   DS 7 3 1080 20091227053717 (
                                    20091127053717 48272 example.jp.
                                    q909fN59Z+Xplt7vfcRXOOm4aVy4mjBju1G5
                                    5vRqco5Hp1BAmqZXGwYRTwbFDl4/BKCL6EiG
                                    w2+egWzEYc/olg== )
```

確認項目 A-85. セキュリティ対応権威サーバはEDNS0によるUDP通信が可能であること

セキュリティ対応権威サーバは EDNS0 による UDP 通信が可能であること。
(問い合わせに対するサーバの応答結果が 512 バイトを超えても、EDNS0 による通信が可能であり、クライアントが正常に応答を受け取れること。

確認項目 A-86. セキュリティ対応権威サーバは 1220 バイトの UDP メッセージをサポートしていること

セキュリティ対応権威サーバは 1220 バイトの UDP メッセージをサポートしていること。
(問い合わせに対するサーバの応答結果が 1220 バイトを超えても、EDNS0 による通信が可能であり、クライアントが正常に応答を受け取れること。

確認項目 A-87. セキュリティ対応権威サーバは 4000 バイトの UDP メッセージをサポートすべき

セキュリティ対応権威サーバは 4000 バイトの UDP メッセージをサポートすべき。
(問い合わせに対するサーバの応答結果が 4000 バイトを超えても、EDNS0 による通信が可能であり、クライアントが正常に応答を受け取れること。

前提事項：

・ DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みとしていること。

確認方法１：

応答結果が 512 バイトを超えるような問い合わせを権威サーバに対して行う。

※ このような状況の便利な作り方：
dnssec-keygen コマンドを何度か実行し、鍵ファイルをいくつか作成する。作成した鍵ファイルをゾーンファイルに追記し、dnssec-signzone コマンドで署名することで、ゾーン名に対する複数の DNSKEY レコード、またそれぞれに対応する RRSIG レコードが作成される。
このようなゾーンファイルを作成し、権威サーバにゾーン名の DNSKEY レコードを問

い合わせることで、作成した分の DNSKEY レコードと RRSIG レコードが応答に含まれるため、データ量を増やすことができる。

dig コマンドに +dnssec および +norec オプションをつけて問い合わせを行う。このとき、以下を指定する。

- @の横には、権威サーバを指定する。
- 権威あるゾーン名を指定する。
  (下記の例では、example.jp としている)
- レコードタイプは DNSKEY を指定する。

```
$ dig +dnssec +norec @192.0.2.1 example.jp. DNSKEY

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 example.jp. DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27840
;; flags: qr aa; QUERY: 1, ANSWER: 10, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      DNSKEY

;; ANSWER SECTION:
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcMbORS8UoU2uVH3D1Bf09jmlGqLOx1
f6FwV9/H9pcuVrypOg4tR pXDqpemc9uYBtF3OxuHAe/xEis/FsV7LFz8=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcOrMNzTadDMUxqfOoXOzCMETFwFboO
sBjRTwf31wdXCTv5z6nBX 2JkQE83PmdKooPLRNdyNUvs8blaUAXUeEos=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAeWVxTfOuNyChnZqOja9tWef/69J4T4
MmkRWEQrO37LggfDiuWL3 eoEb3MxXxFgNqmXXZ26VmQXyflkb1NZCyZs=
example.jp.             1080    IN      DNSKEY  257 3 7 AwEAAdGObAiOdHfblc4u/lTk/41LmSQvNDE
GybTsWqr4xZEnPvqnhZio ZBK6Au2doBLJE11p4lXuuqFQH1UPYGd5EPE=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAagFpChiBjROKURQfqVgEl87BJS/Ox8
vrtal9MSg6uY8F5o56usv O9w99spOUUdlk4/VRoRCl/PMduZROt5rnqU=
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 2105 example.jp. hPAddQ9p8rO3/HE8k8NPRkBQQeNAcYOeotc+DrdddjONwoFLX9soOHVe WE23CQoEF4Ruen
MLDKK42MiirgM9ug==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 14730 example.jp. JlPl1rF+OslBONmg5ZlBP1Qw1+rp/54nHLnfR/e89xvOeXtzRXJp9eLs d1EPOQgENYekd
m8L3SG5mx/a3VwrJg==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 33692 example.jp. W9q5fxgPOwJ+1ki8K91jttfvCb68eK6cghhMShD2AwmsNVMB6zUoOOy2 RnKmOAAqAnbiC
NBZKWXXl7EOtmOemA==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 44432 example.jp. AtqfXqsTrUhe2MwqxBmOsQdP1lyLTmMcN+BFH9KKUzKHP5Yzu7T88Aj+ OuzTEKwpCDKBN
SzgkK7RlAdHBjlmXA==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 60458 example.jp. WsajWVtOAajZZL3ZflrSNEXyzFXc8elalzS7trlGQhf2/d7B56g1bUDV ekobSs+z8bFut
93rDcqgGnZujmJtBg==

;; Query time: 2 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Tue Dec  1 18:18:34 2009
;; MSG SIZE  rcvd: 989
```

以下の点を確認する。

1. example.jp ゾーンに対するDNSKEYレコードとRRSIGレコードが応答に含まれていること。
2. 問い合わせの応答結果の一番最後にある「MSG SIZE rcvd:」という箇所において、応答結果のデータ量が512を超えていること(この例では989)。
3. digコマンドの応答結果の先頭(コマンドラインでコマンドを入力したすぐ下の行)に、「;; Truncated, retrying in TCP mode.」というメッセージが表示されていないこと。
   これは、権威サーバからの応答が512バイトを超えてもTCPへの切り替わりが起きず、UDPで通信が行われていることを表す。
4. 応答結果のOPT PSEUDOSECTION部において、udp:4096となっていること。
   これは、権威サーバ側の設定において、EDNS0によるUDP通信を行う場合、4096バイトのデータ量までサポートされていることを表す。

ここまで確認ができれば、権威サーバはEDNS0による通信により、512バイトを超えるデータ量であってもUDP通信を行えていることになる。

トラブルシューティング１：

1. 問い合わせの応答結果の一番最後にある「MSG SIZE rcvd:」という箇所において、応答結果のデータ量が512を超えていない

   原因:
   問い合わせのデータ量が512バイトに足りていないので、EDNS0の動作検証としては適していない。
   以下の点を確認する。
   - ゾーンファイルに設定するデータを確認し、512バイトを超えるような応答が得られるデータを作成する。

2. digコマンドの応答結果の先頭に「;; Truncated, retrying in TCP mode.」というメッセージが表示されている。
   また、応答結果のOPT PSEUDOSECTION部において、udp:の箇所が4096より小さい数字となっている(下記の例では512)。

```
$ dig +dnssec +norec @192.0.2.1 example.jp. DNSKEY
;; Truncated, retrying in TCP mode.

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 example.jp. DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6997
;; flags: qr aa; QUERY: 1, ANSWER: 10, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;example.jp.                    IN      DNSKEY

;; ANSWER SECTION:
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAeWVxTfOuNyChnZqOja9tWef/69J4T4
MmkRWEQrO37LggfDiuWL3 eoEb3MxXxFgNqmXXZ26VmQXyflkb1NZCyZs=
example.jp.             1080    IN      DNSKEY  257 3 7 AwEAAdGObAiOdHfblc4u/lTk/41LmSQvNDE
GybTsWqr4xZEnPvqnhZio ZBK6Au2doBLJE11p4lXuuqFQH1UPYGd5EPE=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAagFpChiBjROKURQfqVgEl87BJS/Ox8
vrtal9MSg6uY8F5o56usv O9w99spOUUdlk4/VRoRCl/PMduZROt5rnqU=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcMbORS8UoU2uVH3D1Bf09jmlGqLOx1
f6FwV9/H9pcuVrypOg4tR pXDqpemc9uYBtF3OxuHAe/xEis/FsV7LFz8=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcOrMNzTadDMUxqfOoXOzCMETFwFboO
sBjRTwf31wdXCTv5z6nBX 2JkQE83PmdKooPLRNdyNUvs8blaUAXUeEos=
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 2105 example.jp. hPAddQ9p8rO3/HE8k8NPRkBQQeNAcYOeotc+DrdddjONwoFLX9soOHVe WE23CQoEF4Ruen
MLDKK42MiirgM9ug==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 14730 example.jp. JlPl1rF+OslBONmg5ZlBP1Qw1+rp/54nHLnfR/e89xvOeXtzRXJp9eLs d1EPOQgENYekd
m8L3SG5mx/a3VwrJg==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 33692 example.jp. W9q5fxgPOwJ+1ki8K91jttfvCb68eK6cghhMShD2AwmsNVMB6zUoOOy2 RnKmOAAqAnbiC
NBZKWXXl7EOtmOemA==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 44432 example.jp. AtqfXqsTrUhe2MwqxBmOsQdP1lyLTmMcN+BFH9KKUzKHP5Yzu7T88Aj+ OuzTEKwpCDKBN
SzgkK7RlAdHBjlmXA==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 60458 example.jp. WsajWVtOAajZZL3ZflrSNEXyzFXc8elalzS7trlGQhf2/d7B56g1bUDV ekobSs+z8bFut
93rDcqgGnZujmJtBg==

;; Query time: 3 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Tue Dec  1 20:07:07 2009
;; MSG SIZE  rcvd: 989
```

原因:

権威サーバ側で、EDNS0 による UDP 通信の最大データ量が制限されている可能性がある(この例では 512 バイトに絞られている)。

この例での問い合わせのデータ量は 989 バイトだが、UDP 通信での最大データ量がこれより小さいため、TCP への切り替わりが起きている。

以下の点を確認する。

- BIND の named.conf の options ブロックにおいて、以下のような 2 行が設定されていないか確認する(edns-udp-size 文と max-udp-size 文)。

```
edns-udp-size 512;
max-udp-size 512;
```

この数字を 512 より大きな数字に上げるか、edns-udp-size 文と max-udp-size 文を削除する。

named.conf に edns-udp-size 文と max-udp-size 文を明示的に記述しない場合、4096 が設定される。

3. dig コマンドの応答が失敗する。

```
$ dig +dnssec +norec @192.0.2.1 example.jp. DNSKEY
;; Truncated, retrying in TCP mode.
;; Connection to 192.0.2.1#53(192.0.2.1) for example.jp. failed: host unreachable.
$
```

原因:

権威サーバ側で、EDNS0 による UDP 通信の最大データ量が制限されている可能性があり、TCP への切り替わりが起きている。

しかしクライアントと権威サーバとの間の TCP での通信がブロックされている可能性がある。

以下の点を確認する。

- クライアントと権威サーバとの間で、TCP、あて先ポートが 53 番の通信がブロックされていないか確認する。ブロックされている場合、開放する必要がある(TCP でも通信できることが必須事項)。

  ブロックされているかどうかの確認方法については、以下を参照のこと。

  確認項目 共通-1. TCP の通信がブロックされていないことの確認

確認方法 2.

次に、今回と同じ問い合わせを+bufsize=512 オプションを付加して行う。

このオプションは、クライアント側から EDNS0 UDP 通信での最大バイトサイズを 512 に制限することを権威サーバに通知する。

※ dig コマンドの場合、+bufsize= オプションを付加しない場合はデフォルトで 65535 が適用される。

```
$ dig +dnssec +norec +bufsize=512 @192.0.2.1 example.jp. DNSKEY
;; Truncated, retrying in TCP mode.

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec +bufsize=512 @192.0.2.1 example.jp. DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26628
;; flags: qr aa; QUERY: 1, ANSWER: 10, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      DNSKEY

;; ANSWER SECTION:
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcMbORS8UoU2uVH3D1Bf09jmlGqLOx1
f6FwV9/H9pcuVrypOg4tR pXDqpemc9uYBtF3OxuHAe/xEis/FsV7LFz8=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcOrMNzTadDMUxqfOoXOzCMETFwFboO
sBjRTwf31wdXCTv5z6nBX 2JkQE83PmdKooPLRNdyNUvs8blaUAXUeEos=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAeWVxTfOuNyChnZqOja9tWef/69J4T4
MmkRWEQrO37LggfDiuWL3 eoEb3MxXxFgNqmXXZ26VmQXyflkb1NZCyZs=
example.jp.             1080    IN      DNSKEY  257 3 7 AwEAAdGObAiOdHfblc4u/lTk/41LmSQvNDE
GybTsWqr4xZEnPvqnhZio ZBK6Au2doBLJE11p4lXuuqFQH1UPYGd5EPE=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAagFpChiBjROKURQfqVgEl87BJS/Ox8
vrtal9MSg6uY8F5o56usv O9w99spOUUdlk4/VRoRCl/PMduZROt5rnqU=
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 2105 example.jp. hPAddQ9p8rO3/HE8k8NPRkBQQeNAcYOeotc+DrdddjONwoFLX9soOHVe WE23CQoEF4Ruen
MLDKK42MiirgM9ug==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 14730 example.jp. JlPl1rF+OslBONmg5ZlBP1Qw1+rp/54nHLnfR/e89xvOeXtzRXJp9eLs d1EPOQgENYekd
m8L3SG5mx/a3VwrJg==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 33692 example.jp. W9q5fxgPOwJ+1ki8K91jttfvCb68eK6cghhMShD2AwmsNVMB6zUoOOy2 RnKmOAAqAnbiC
NBZKWXXl7EOtmOemA==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 44432 example.jp. AtqfXqsTrUhe2MwqxBmOsQdP1lyLTmMcN+BFH9KKUzKHP5Yzu7T88Aj+ OuzTEKwpCDKBN
SzgkK7RlAdHBjlmXA==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 60458 example.jp. WsajWVtOAajZZL3ZflrSNEXyzFXc8elalzS7trlGQhf2/d7B56g1bUDV ekobSs+z8bFut
93rDcqgGnZujmJtBg==

;; Query time: 1 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Tue Dec  1 18:55:46 2009
;; MSG SIZE  rcvd: 989
```

以下の点を確認する。

1. dig コマンドの応答結果の先頭(コマンドラインでコマンドを入力したすぐ下の行)に、「;; Truncated, retrying in TCP mode.」というメッセージが表示されていること。
   これは、問い合わせに対するサーバの応答のデータ量(989)がクライアント側から指定した UDP の最大データサイズを超えたため、TCP でリトライして通信が行われたことを表す。
   この例では TCP でリトライして通信が行われているものの、問い合わせの結果は正常に得られているので、TCP への切り替えが正常に行われていることになる。

トラブルシューティング 2：

1. dig コマンドの応答が失敗する。
   上記トラブルシューティング 1: の 3. と同様、クライアントと権威サーバとの間で TCP の通信がブロックされている可能性がある。
   上記トラブルシューティング 1: の 3.と同様の確認を行う。

確認方法 3：

応答結果が 1220 バイトを超えるような問い合わせを権威サーバに対して行う。

```
$ dig +dnssec +norec @192.0.2.1 example.jp DNSKEY

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.1 example.jp DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42803
;; flags: qr aa; QUERY: 1, ANSWER: 30, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      DNSKEY

;; ANSWER SECTION:
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcMbORS8UoU2uVH3D1Bf09jmlGqLOx1
f6FwV9/H9pcuVrypOg4tR pXDqpemc9uYBtF3OxuHAe/xEis/FsV7LFz8=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcOrMNzTadDMUxqfOoXOzCMETFwFboO
sBjRTwf31wdXCTv5z6nBX 2JkQE83PmdKooPLRNdyNUvs8blaUAXUeEos=
～略～
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20100101020006 200912020200
06 60458 example.jp. Fa+LRGd/MNg831wlO9557MosuRhzs2EK9mxeNEDNyxDHCTQlZHj3aYxt roSh5fZNqHx10
CUkqBZ9VhcMkHOknA==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20100101020006 200912020200
06 65309 example.jp. BLALrwtCjpTNUJVZRjdr1HgOzwcxONDL/lowMoxkMkAlxVQjsi3cfCEt ODge1sVcZVTKl
w4hTUmXYLRXR6zwOA==

;; Query time: 28 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Wed Dec  2 15:58:18 2009
;; MSG SIZE  rcvd: 2889
```

以下の点を確認する。

1. example.jp ゾーンに対する DNSKEY レコードと RRSIG レコードが応答に含まれていること。
2. 問い合わせの応答結果の一番最後にある「MSG SIZE rcvd:」という箇所において、応答結果のデータ量が 1220 を超えていること(この例では 2889)。
3. dig コマンドの応答結果の先頭(コマンドラインでコマンドを入力したすぐ下の行)に、

「;; Truncated, retrying in TCP mode.」というメッセージが表示されていないこと。

トラブルシューティング 3：

1. dig コマンドの応答結果の先頭に「;; Truncated, retrying in TCP mode.」というメッセージが表示されている。

   原因:
   権威サーバ側で、EDNS0 による UDP 通信の最大データ量が制限されている可能性がある。
   UDP 通信での最大データ量が小さいため、TCP への切り替わりが起きている。
   以下の点を確認する。
   - 上記トラブルシューティング 1. の 2.を参照し、権威サーバ側の設定で、EDNS0 による UDP 通信の最大データ量が制限されていないか確認する。
     制限をかける場合であっても、少なくとも 1220 より大きな値を設定する。

2. dig コマンドの応答が失敗する。

   原因:
   権威サーバからの応答のデータ量がクライアントと権威サーバの間の通信路の MTU を越えており、パケットのフラグメントが起きている可能性がある。
   権威サーバ側の設定において、EDNS0 による UDP 通信の最大データ量を 1220 より大きくしていても、通信路の MTU がこれより小さければ、パケットのフラグメントが発生する。
   また、通信路においてフラグメントが起きたパケットを落としている可能性がある。
   以下の点を確認してください。
   - クライアントと権威サーバの間の通信路の MTU が、1220 を越えているかどうか。
     1220 より小さい場合、可能であれば、通信路のネットワーク機器の設定を変更し、MTU の値を大きくする。
   - クライアントと権威サーバの間の通信路において、フラグメントしているパケットを落とすような設定がされていないか。
     可能であれば、通信路のネットワーク機器の設定を変更し、パケットがフラグメントしていても通すようにする。

確認方法 4：

応答結果が 4000 バイトを超えるような問い合わせを権威サーバに対して行う。

応答結果が 4000 バイトを超えるような問い合わせであっても、正常に応答結果が得られることを確認する。

確認項目 A-95. セキュリティ対応権威サーバは DO=1 の問い合わせに応答する場合、RRSIG レコードが応答に含まれることの確認

署名付きゾーンを持つ権威サーバは、リゾルバから DO=1(DO ビットが立っている)となっている問い合わせに応答する場合、権威を持つリソースレコードに対する応答には RRSIG レコードが含まれていること。

前提事項：

- DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みとしていること。

確認方法：

権威サーバに対し、署名したゾーンに設定してあるはずのリソースレコードを確認する。
ここでは例として www.example.jp の A レコードを確認する。
権威サーバに対して、以下の問い合わせを行う。

dig コマンドに +dnssec および +norec オプションをつけて問い合わせを行う(+dnssec オプションをつけることで、リゾルバから DO=1 の問い合わせを行うことになる)。このとき、以下を指定する。

- @の横には権威サーバを指定する。
- 署名したゾーンに設定してあるはずのリソースレコードのオーナ名を指定する。
  (下記の例では、www.example.jp としている)
- レコードタイプは A を指定する。

```
$ dig +dnssec +norec @192.0.2.1 www.example.jp A

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 www.example.jp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26665
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;pc1.example.jp.                        IN      A

;; ANSWER SECTION:
www.example.jp.         1080    IN      A       192.0.2.2
www.example.jp.         1080    IN      RRSIG   A 7 3 1080 20091227053717 20091127053717 4
8272 example.jp. i6xlKiaVfxtAWptexxC8z9MxzpwPpbJuc7U+rMLnNxUzlq91lOWCDNU1 ttoKfyYqy9cRm+wd
Uz2AXtqh9lRIVg==

;; AUTHORITY SECTION:
example.jp.             1080    IN      NS      ns.example.jp.
example.jp.             1080    IN      RRSIG   NS 7 2 1080 20091227053717 20091127053717 4
8272 example.jp. MTdauvE1dMaHZE/litFXRSUZFZU+v78oHEhLKcscecjBKCwei9qsNB6X +By2eDolcwkYPH9PF
pzYaRGTqUyeDA==

;; ADDITIONAL SECTION:
ns.example.jp.          1080    IN      A       192.0.2.1
ns.example.jp.          1080    IN      RRSIG   A 7 3 1080 20091227053717 20091127053717 482
72 example.jp. aJcHslgP6nd78Ym5MuiMNRYttcb/6yloiCtBhT/W+kMJklozhQ1MG6NN sAB9xbrPjnjOFbYkr/Q
A6z49GDTtKw==

;; Query time: 2 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Mon Nov 30 18:04:42 2009
;; MSG SIZE  rcvd: 411
```

得られたレスポンスの OPT PSEUDOSECTION 部において、do ビットが立っていることを確認する。
また、得られたレスポンスに、問い合わせたオーナ名(この例では www.example.jp)に対する A レコードが含まれていることを確認する。
また、A レコードに対する RRSIG レコードが応答に含まれていることを確認する。具体的には以下の点を確認する。

- オーナ名が www.example.jp となっている RRSIG レコードが応答に含まれていること
- その RRSIG レコードの署名対象部(RRSIG のすぐ右)が A となっていること

トラブルシューティング：

1. 問い合わせたホスト名に対する A レコードが応答に含まれていない

原因:
問い合わせたオーナ名が間違っているか、権威サーバが持つ権威あるゾーンの中に、そのオーナ名が設定されていない。
以下の点を確認する。

- BIND の named.conf で指定しているゾーンファイルに、そのオーナ名の A レコードが設定されていること。

2. 得られた応答に do ビットが立っていない

原因:
問い合わせ時に +dnssec オプションをつけていない場合、RRSIG レコードは権威サーバからの応答に含まれない。
dig コマンド実行時に、+dnssec オプションをつけていることを確認する。

3. 得られた応答に A レコードに対する RRSIG レコードが含まれていない

原因 1:
権威サーバにおいて、DNSSEC が無効にされていないことを確認する。

以下を参照し、DNSSEC が有効になっていることを確認する。
確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認

原因 2:
BIND が読み込んでいるゾーンファイルが署名前のゾーンファイルとなっている

以下の点を確認する。

- ゾーンの署名をしたが、そのファイルがまだ BIND に読み込まれていない。あるいは、BIND の named.conf で指定しているゾーンファイルがまちがっている。
  以下を参照し、署名したゾーンファイルが BIND に読み込まれていることを確認する。
  確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認

確認項目 A-115. セキュリティ対応権威サーバは委任点の参照を応答する場合、DS とその RRSIG レコードが応答の権威部に含まれることの確認

DO=1 の問い合わせに対し、署名付きゾーンを持つ権威サーバが委任先の参照を返す場合、委任点に DS が存在する場合は、NS に加え、DS とその RRSIG を共に権威部に付加しなければならない。

前提事項：

- DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みとしていること。
- 署名付きゾーンの DS レコードを、権威サーバに登録済みであること。

ドメインの構成：

- 構築した DNSSEC 対応の権威サーバ(ここでは「構築済みの権威サーバ」と呼ぶ)がもつ、署名付きの権威あるゾーン名を、example.jp とする。
- 委任先のゾーン名を、sub.example.jp とする。
- sub.example.jp は、example.jp の子ゾーンになる。

確認方法：

権威サーバに、あらかじめ登録済みである sub.example.jp の NS レコードを確認する。
権威サーバに対して、以下の問い合わせを行う。

dig コマンドに +dnssec および +norec オプションをつけて問い合わせを行う。このとき、以下を指定する。
- @の横には権威サーバを指定する。
- 委任先のゾーン名を指定する。この例では sub.example.jp となる。

レコードタイプは NS を指定する。

```
$ dig +dnssec +norec @192.0.2.1 sub.example.jp NS

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 sub.example.jp NS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57924
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      NS

;; AUTHORITY SECTION:
sub.example.jp.         1080    IN      NS      ns.sub.example.jp.
sub.example.jp.         1080    IN      DS      23454 7 2 B49C5A8AE492A44BBDA45908E114FB5C
79AEF74F4BE7E5FEDB5312FF C120ACF3
sub.example.jp.         1080    IN      DS      23454 7 1 2D35FC7B0197A150A8D8958964C98325
9EC271E1
sub.example.jp.         1080    IN      RRSIG   DS 7 3 1080 20091227053717 20091127053717
48272 example.jp. q909fN59Z+Xplt7vfcRX0Om4aVy4mjBju1G55vRqco5Hp1BAmqZXGwYR TwbFDl4/BKCL6Ei
Gw2+egWzEYc/olg==

;; ADDITIONAL SECTION:
ns.sub.example.jp.      1080    IN      A       192.0.2.101

;; Query time: 2 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Mon Nov 30 19:40:28 2009
;; MSG SIZE  rcvd: 266
```

得られたレスポンスの OPT PSEUDOSECTION 部において、do ビットが立っていることを確認する。
また、得られたレスポンスの AUTHORITY SECTION に、委任先のゾーン名(この例では sub.example.jp)の NS レコードが含まれていることを確認する。
また、得られたレスポンスの AUTHORITY SECTION に、委任先のゾーン名の DS レコードが含まれていることを確認する。
また、得られたレスポンスの AUTHORITY SECTION に、DS レコードに対する RRSIG レコードが含まれていることを確認する。具体的には以下の点を確認する。

- オーナ名が sub.example.jp となっている RRSIG レコードが応答に含まれていること
- その RRSIG レコードの署名対象部(RRSIG のすぐ右)が DS となっていること

トラブルシューティング：

1. 問い合わせた委任先のゾーン名に対する NS レコードが応答に含まれていない

   原因:

問い合わせたゾーン名が間違っているか、権威サーバが持つ権威あるゾーンの中に、そのゾーン名の NS レコードが設定されていない。
以下の点を確認する。

- BIND の named.conf で指定しているゾーンファイルに、そのゾーン名の NS レコードが設定されていること。

2. 得られた応答に do ビットが立っていない

原因:
問い合わせ時に +dnssec オプションをつけていない場合、RRSIG レコードは権威サーバからの応答に含まれない。
dig コマンド実行時に、+dnssec オプションをつけていることを確認する。

3. 得られた応答の AUTHORITY SECTION に DS レコードが含まれていない

原因 1:
権威サーバにおいて、DNSSEC が無効にされていないことを確認する。

以下を参照し、DNSSEC が有効になっていることを確認する。
確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認

原因 2:
権威サーバのゾーンファイルに、委任先のゾーン名の DS レコードが設定されていない。

委任先のゾーン名の DS レコードがないため、委任先ゾーンは署名されないゾーンとなっている。
以下の点を確認する。

- BIND が読みこんでいる署名後のゾーンファイルに、委任先のゾーン名の DS レコードが設定されていること。
- 下記のように、BIND が読みこんでいるゾーンファイルが署名後のものとなっていること。

4. 得られた応答に、DS レコードに対する RRSIG レコードが含まれていない

原因 1:

BIND が読みこんでいるゾーンファイルが、署名前のゾーンファイルとなっている

以下を参照し、署名したゾーンファイルが BIND に読み込まれていることを確認する。
確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認

確認項目 A-229. ゾーンの頂点に NSEC3PARAM レコードが存在すること。

署名したゾーンを持つ権威サーバは、NSEC3PARAM レコードを該当ゾーンに保持していること。

前提事項：

- DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みとしていること。
- DNSSEC の不在証明が NSEC3 レコード形式で行われていること。

確認方法：

構築済みの権威サーバに対して、以下の問い合わせを行う。

dig コマンドに +dnssec および +norec オプションをつけて問い合わせを行う。このとき、以下を指定する。

- @の横には、権威サーバを指定する。
- 権威あるゾーン名を指定する。
  (下記の例では、example.jp としている)
- レコードタイプは NSEC3PARAM を指定する。

```
$ dig +dnssec +norec @192.0.2.1 example.jp NSEC3PARAM

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 example.jp NSEC3PARAM
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16263
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      NSEC3PARAM

;; ANSWER SECTION:
example.jp.             0       IN      NSEC3PARAM 1 0 100 AAAA
example.jp.             0       IN      RRSIG   NSEC3PARAM 7 2 0 20091227053717 2009112705
3717 48272 example.jp. HQW/cli29v3rYHOZqvt/kuQTWSWG2Rx+LwloSzRoOAHu7GEfLTQ+adaC HLQF3vJRn+
Q3U4DUa8MauelJuc+b+g==

;; AUTHORITY SECTION:
example.jp.             1080    IN      NS      ns.example.jp.
example.jp.             1080    IN      RRSIG   NS 7 2 1080 20091227053717 20091127053717 4
8272 example.jp. MTdauvE1dMaHZE/litFXRSUZFZU+v78oHEhLKcscecjBKCwei9qsNB6X +By2eDolcwkYPH9PF
pzYaRGTqUyeDA==

;; ADDITIONAL SECTION:
ns.example.jp.          1080    IN      A       192.0.2.1
ns.example.jp.          1080    IN      RRSIG   A 7 3 1080 20091227053717 20091127053717 482
72 example.jp. aJcHslgP6nd78Ym5MuiMNRYttcb/6yloiCtBhT/W+kMJklozhQ1MG6NN sAB9xbrPjnjOFbYkr/Q
A6z49GDTtKw==

;; Query time: 1 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Fri Nov 27 21:33:52 2009
;; MSG SIZE  rcvd: 410
```

得られたレスポンスのANSWERセクションにNSEC3PARAMレコードが含まれていることを確認する。

トラブルシューティング：

1. NSEC3PARAMレコードが含まれていない

   原因:
   権威サーバが正しくDNSSEC対応として設定されていない。
   以下の点を確認する。

   - BINDのnamed.confで指定しているゾーンファイルはdnssec-signzone コマンドでゾーンファイルを署名したものであり、署名したゾーンファイルにNSEC3PARAMレコードが含まれていること。

例：署名したゾーンファイルに、以下のような行が含まれていること。

```
0        NSEC3PARAM 1 0 100 AAAA
```

もし含まれていない場合、dnssec-signzone コマンドによる署名が正しく行えていない。

署名時に NSEC3PARAM レコードを生成させるには、dnssec-signzone コマンド実行時に、-3 <SALT 値> のオプションを指定することが必要である。

-3 オプションを指定せずに dnssec-signzone コマンドを実行した場合、NSEC 形式となり、NSEC3PARAM レコードは生成されない。

確認項目 A-246. 非 opt-out 運用のゾーンで opt-out なしの NSEC3、またそれに対する RRSIG が返却されることの確認

非 opt-out 運用のゾーンに対する署名されない子ゾーンの問い合わせに応答する際に、委任名に一致する NSEC3 レコードが存在する場合、NSEC3 レコードが応答に含まれること。また NSEC3 レコードの bit map に DS は存在しないこと。また、NSEC3 レコードに対する RRSIG レコードが応答に含まれていること。

前提事項：

- DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みとしていること。
- 署名されない子ゾーンの NS レコードを親側の権威サーバに登録済みであること

ドメインの構成：

- 構築した DNSSEC 対応の権威サーバ(ここでは「構築済みの権威サーバ」と呼ぶ)がもつ、署名付きの権威あるゾーン名を example.jp とする。
- 署名されない子ゾーン名を sub.example.jp とする。
- sub.example.jp は、example.jp の子ゾーンになる。

確認方法：

構築済みの権威サーバに対して、以下の問い合わせを行う。

dig コマンドに +dnssec および+norec オプションをつけて問い合わせを行う。このとき、以下を指定する。

- @の横には、権威サーバを指定する。
- 署名されない子ゾーン名を指定する。
  (下記の例では、sub.example.jp としている)
- レコードタイプは NS を指定する。

```
$ dig +dnssec +norec @192.0.2.1 sub.example.jp  NS

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 sub.example.jp NS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2029
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      NS

;; AUTHORITY SECTION:
sub.example.jp.         1080    IN      NS      ns.sub.example.jp.
MEDOGC5QG5DRTEF55B22726N6PDN1PMQ.example.jp. 8640 IN NSEC3 1 0 100 AAAA MN1GSBAJG3NPUPRGE0
FA5G8I2PD2UNE4 NS
MEDOGC5QG5DRTEF55B22726N6PDN1PMQ.example.jp. 8640 IN RRSIG NSEC3 7 3 8640 20091230034908 2
0091130034908 48272 example.jp. e2q1nXZRPUDJCwMEv44k47rl2zLuyKOp9RkX9k/+CNP+vmVhJL5ytxgd 5
bdiiKIsu7aBXDIoAnZlHevSx+qoCA==

;; ADDITIONAL SECTION:
ns.sub.example.jp.      1080    IN      A       192.0.2.101

;; Query time: 1 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Mon Nov 30 13:57:17 2009
;; MSG SIZE  rcvd: 258
```

得られたレスポンスに、署名されない子ゾーンの NS レコードが含まれていることを確認する。
また、署名されない子ゾーン名に対する NSEC3 レコードが 1 件応答に含まれることを確認する。
また、得られた NSEC3 レコードのフラグフィールド部(NSEC3 の 2 つ右)の数値が 0 であることを確認する。0 は「opt-put なし」であることを意味する。
また、得られた NSEC3 レコードの bit map には DS が含まれていないことを確認する（上記の例では NS しか含まれていない)。
また、上記の NSEC3 レコードに対する RRSIG レコードが応答に含まれていることを確認する。具体的には以下の点を確認する。

- オーナ名が、上記 NSEC3 レコードと同じである RRSIG レコードが応答に含まれていること
- その RRSIG レコードの署名対象部(RRSIG のすぐ右)が NSEC3 となっていること

トラブルシューティング：

1. 署名されない子ゾーン名の NS レコードが応答に含まれていない

原因:
親側の権威サーバにて、子ゾーンの委任点が正しく設定されていない。
以下の点を確認する。

- BIND の named.conf で指定しているゾーンファイルに、署名されない子ゾーンの NS レコードが設定されていること。

2. 署名されない子ゾーン名の NSEC3 レコードが応答に含まれない

原因:
下記の実行結果のように、応答結果に NSEC3 レコードが含まれず、DS レコード(と RRSIG レコード)が含まれている場合は、子ゾーンは署名された状態として扱われている。

```
$ dig +dnssec +norec @192.0.2.1 sub.example.jp NS

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 sub.example.jp NS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30419
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      NS

;; AUTHORITY SECTION:
sub.example.jp.         1080    IN      NS      ns.sub.example.jp.
sub.example.jp.         1080    IN      DS      23454 7 2 B49C5A8AE492A44BBDA45908E114FB5C7
9AEF74F4BE7E5FEDB5312FF C120ACF3
sub.example.jp.         1080    IN      DS      23454 7 1 2D35FC7B0197A150A8D8958964C983259
EC271E1
sub.example.jp.         1080    IN      RRSIG   DS 7 3 1080 20091227053717 20091127053717 4
8272 example.jp. q9O9fN59Z+Xplt7vfcRXOOm4aVy4mjBju1G55vRqco5Hp1BAmqZXGwYR TwbFDl4/BKCL6EiGw
2+egWzEYc/olg==

;; ADDITIONAL SECTION:
ns.sub.example.jp.      1080    IN      A       192.0.2.101

;; Query time: 1 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Mon Nov 30 15:56:59 2009
;; MSG SIZE  rcvd: 266
```

子ゾーンを署名されないゾーンとして運用するのが正しいのであれば、以下の点を確認する。

- 署名前のゾーンファイルに子ゾーン名の DS レコードが記述されていると思われるので、DS レコードの記述を削除し、dnssec-signzone コマンドで署名をやり直

し、BIND に再読み込みさせる。

3. NSEC3 レコードに対する RRSIG レコードが応答に含まれていない

原因:
BIND が読み込んでいるゾーンファイルに、RRSIG レコードが設定されていない。
ゾーンファイルの署名が正しく行われていないことが考えられる。
以下の点を確認する。

- dnssec-signzone コマンドが正しく実行されれば、署名されない子ゾーンの DS レコードがないことの証明(NSEC3 レコードと、対応する RRSIG レコード)が署名後のゾーンファイルに反映されるので、もう一度 dnssec-signzone コマンドによるゾーンの署名をやり直す。

4. 署名されない子ゾーン名の NSEC3 レコードが、複数件応答に含まれる

原因:
下記の実行結果のように、応答結果に NSEC3 レコードが複数件含まれている場合や、NSEC3 レコードのフラグフィールド部が 1 となっている場合は、子ゾーンは署名されない状態となっているものの、opt-out 運用となっている。
NSEC3 レコードのフラグフィールド部が 1 の場合は「opt-out あり」を意味する。

```
$ dig +dnssec +norec @192.0.2.1 sub.example.jp  NS

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 sub.example.jp NS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2510
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 5, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      NS

;; AUTHORITY SECTION:
sub.example.jp.         1080    IN      NS      ns.sub.example.jp.
RI01VRHFG77293BPOLRE6N3OTT7CQGCH.example.jp. 8640 IN NSEC3 1 1 100 AAAA 43F1JQ1J79FOLLN94Q
EV47LHJKMR5CFU NS SOA RRSIG DNSKEY NSEC3PARAM
RI01VRHFG77293BPOLRE6N3OTT7CQGCH.example.jp. 8640 IN RRSIG NSEC3 7 3 8640 20091230035148 20
091130035148 48272 example.jp. JwfWmYJKLKznQTqtsBcalx2JN8zKPDiviP5/PWxhiVhmjnOdkd6eJwZO JaF
UZTA2qehieVEDVxuLXgNZwKOxAw==
LA65M10lOKG4A5KL6QUBKML9VK5NN4G4.example.jp. 8640 IN NSEC3 1 1 100 AAAA MN1GSBAJG3NPUPRGEO
FA5G8I2PD2UNE4 A RRSIG
LA65M10lOKG4A5KL6QUBKML9VK5NN4G4.example.jp. 8640 IN RRSIG NSEC3 7 3 8640 20091230035148 20
091130035148 48272 example.jp. G1N+OnjWZvVAkrOhVRf2ZJwG6Tp+JOLaFml+DgFWOOOvlANps0e24VvS 69e
gXe9+yLBWFTFBOM8cMpqkLk+eYQ==

;; ADDITIONAL SECTION:
ns.sub.example.jp.      1080    IN      A       192.0.2.101

;; Query time: 3 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Mon Nov 30 14:02:19 2009
;; MSG SIZE  rcvd: 451
```

子ゾーンを非 opt-out 運用の署名されないゾーンとして運用するのが正しいのであれば、以下の点を確認する。

- ゾーンの署名時に、dnssec-signzone コマンドを実行したときに、-A オプション(opt-out ありの運用)をつけてゾーンファイルが署名されていることが考えられる。dnssec-signzone コマンドに –A を指定せずに実行してゾーンファイルの署名をやり直し、BIND に再読み込みさせる。
- このとき、署名前のゾーンファイルには、署名されない子ゾーン名の DS レコードが設定されていないことを確認する。

確認項目 A-248. opt-out 運用のゾーンで opt-out の NSEC3 レコードが返却されることの確認

署名されない子ゾーンに対する問い合わせに対して、opt-out のケースでは委任名に一致する NSEC3 レコードが存在しない場合がある。この場合は closest provable encloser の証明が応答に含まれていること。この証明の next closer を cover する NSEC3 レコードは opt-out がセットされていること。また、NSEC3 レコードに対する RRSIG レコードが応答に含まれていること。

前提事項：

- DNSSEC 対応の権威サーバを構築済みであること。また権威あるゾーンを署名済みとしていること。
- 署名されない子ゾーンの NS レコードを親側の権威サーバに登録済みであること(子ゾーンの DS レコードが親側の権威サーバに登録されている状態でないこと)

ドメインの構成：

- 構築した DNSSEC 対応の権威サーバ(ここでは「構築済みの権威サーバ」と呼ぶ)がもつ、署名付きの権威あるゾーン名を、example.jp とする。
- 署名されない子ゾーン名を、sub.example.jp とする。
- sub.example.jp は、example.jp の子ゾーンになる。

確認方法：

構築済みの権威サーバに対して、以下の問い合わせを行う。

dig コマンドに +dnssec および +norec オプションをつけて問い合わせを行う。このとき、以下を指定する。

- @の横には、権威サーバを指定する。
- 署名されない子ゾーン名を指定する。
  (下記の例では、sub.example.jp としている)
- レコードタイプは NS を指定する。

```
$ dig +dnssec +norec @192.0.2.1 sub.example.jp  NS

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 sub.example.jp NS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2510
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 5, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      NS

;; AUTHORITY SECTION:
sub.example.jp.         1080    IN      NS      ns.sub.example.jp.
RI01VRHFG77293BPOLRE6N3OTT7CQGCH.example.jp. 8640 IN NSEC3 1 1 100 AAAA 43F1JQ1J79FOLLN94Q
EV47LHJKMR5CFU NS SOA RRSIG DNSKEY NSEC3PARAM
RI01VRHFG77293BPOLRE6N3OTT7CQGCH.example.jp. 8640 IN RRSIG NSEC3 7 3 8640 20091230035148 2
0091130035148 48272 example.jp. JwfWmYJKLKznQTqtsBcalx2JN8zKPDiviP5/PWxhiVhmjnOdkd6eJwZO J
aFUZTA2qehieVEDVxuLXgNZwKOxAw==
LA65M10l0KG4A5KL6QUBKML9VK5NN4G4.example.jp. 8640 IN NSEC3 1 1 100 AAAA MN1GSBAJG3NPUPRGEO
FA5G8I2PD2UNE4 A RRSIG
LA65M10l0KG4A5KL6QUBKML9VK5NN4G4.example.jp. 8640 IN RRSIG NSEC3 7 3 8640 20091230035148 2
0091130035148 48272 example.jp. G1N+OnjWZvVAkrOhVRf2ZJwG6Tp+JOLaFml+DgFWOOOvlANpsOe24VvS 6
9egXe9+yLBWFTFBOM8cMpqkLk+eYQ==

;; ADDITIONAL SECTION:
ns.sub.example.jp.      1080    IN      A       192.0.2.101

;; Query time: 3 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Mon Nov 30 14:02:19 2009
;; MSG SIZE  rcvd: 451
```

得られたレスポンスに、署名されない子ゾーンの NS レコードが含まれていることを確認する。

また、署名されない子ゾーン名に対する NSEC3 レコードが複数件応答に含まれることを確認する。

また、得られた NSEC3 レコードのフラグフィールド部(NSEC3 の 2 つ右)の数値が 1 であることを確認する。1 は「opt-put あり」であることを意味する。

また、上記の NSEC3 レコードに対する RRSIG レコードが応答に含まれていることを確認する。具体的には以下の点を確認する。

- オーナ名が、上記 NSEC3 レコードと同じである RRSIG レコードが応答に含まれていること
- その RRSIG レコードの署名対象部(RRSIG のすぐ右)が NSEC3 となっていること

トラブルシューティング：

1. 署名されない子ゾーン名の NS レコードが応答に含まれていない

   原因:
   親側の権威サーバにて、子ゾーンの委任点が正しく設定されていない。
   以下の点を確認する。

   - BIND の named.conf で指定しているゾーンファイルに、署名されない子ゾーンの NS レコードが設定されていること。

2. 署名されない子ゾーン名の NSEC3 レコードが応答に含まれない。

   原因:
   下記の実行結果のように応答結果に NSEC3 レコードが含まれず、DS レコード(と RRSIG レコード)が含まれている場合は、子ゾーンは署名された状態となっている。

```
$ dig +dnssec +norec @192.0.2.1 sub.example.jp NS

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 sub.example.jp NS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30419
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      NS

;; AUTHORITY SECTION:
sub.example.jp.         1080    IN      NS      ns.sub.example.jp.
sub.example.jp.         1080    IN      DS      23454 7 2 B49C5A8AE492A44BBDA45908E114FB5C7
9AEF74F4BE7E5FEDB5312FF C120ACF3
sub.example.jp.         1080    IN      DS      23454 7 1 2D35FC7B0197A150A8D8958964C983259
EC271E1
sub.example.jp.         1080    IN      RRSIG   DS 7 3 1080 20091227053717 20091127053717 4
8272 example.jp. q909fN59Z+Xplt7vfcRXOOm4aVy4mjBju1G55vRqco5Hp1BAmqZXGwYR TwbFDl4/BKCL6EiGw
2+egWzEYc/olg==

;; ADDITIONAL SECTION:
ns.sub.example.jp.      1080    IN      A       192.0.2.101

;; Query time: 1 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Mon Nov 30 15:56:59 2009
;; MSG SIZE  rcvd: 266
```

   子ゾーンを署名されないゾーンとして運用するのが正しいのであれば、以下の点を確認する。

   - 署名前のゾーンファイルに、子ゾーン名の DS レコードが記述されていると思われるので、DS レコードの記述を削除し、dnssec-signzone コマンドで署名をやり直

し、BIND に再読み込みさせる。

3. NSEC3 レコードに対する RRSIG レコードが応答に含まれていない

原因:
BIND が読み込んでいるゾーンファイルに、RRSIG レコードが設定されていない。
ゾーンファイルの署名が正しく行われていないことが考えられる。
以下の点を確認する。

- dnssec-signzone コマンドを実行すれば、署名されない子ゾーンの DS レコードがないことの証明となるレコード(NSEC3 レコードと、対応する RRSIG レコード)が署名後のゾーンファイルに生成されるので、もう一度 dnssec-signzone コマンドによるゾーンの署名をやり直す。

4. 署名されない子ゾーン名の NSEC3 レコードが、1 件のみ応答に含まれる(委任名に一致する NSEC3 レコードが存在している)

原因:
下記の実行結果のように、応答結果に NSEC3 レコードが 1 件含まれている場合や、NSEC3 レコードのフラグフィールド部が 0 となっている場合は、子ゾーンは署名されない状態となっているものの、非 opt-out 運用となっている。
NSEC3 レコードのフラグフィールド部が 0 の場合は「opt-out あり」を意味する。

```
$ dig +dnssec +norec @192.0.2.1 sub.example.jp  NS

; <<>> DiG 9.6.1-P1 <<>> +dnssec +norec @192.0.2.1 sub.example.jp NS
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2029
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      NS

;; AUTHORITY SECTION:
sub.example.jp.         1080    IN      NS      ns.sub.example.jp.
MEDOGC5QG5DRTEF55B22726N6PDN1PMQ.example.jp. 8640 IN NSEC3 1 0 100 AAAA MN1GSBAJG3NPUPRGE0
FA5G8I2PD2UNE4 NS
MEDOGC5QG5DRTEF55B22726N6PDN1PMQ.example.jp. 8640 IN RRSIG NSEC3 7 3 8640 20091230034908 20
091130034908 48272 example.jp. e2q1nXZRPUDJCwMEv44k47rl2zLuyKOp9RkX9k/+CNP+vmVhJL5ytxgd 5bd
iiKIsu7aBXDIoAnZlHevSx+qoCA==

;; ADDITIONAL SECTION:
ns.sub.example.jp.      1080    IN      A       192.0.2.101

;; Query time: 1 msec
;; SERVER: 192.0.2.1#53(192.0.2.1)
;; WHEN: Mon Nov 30 13:57:17 2009
;; MSG SIZE  rcvd: 258
```

子ゾーンを opt-out 運用の署名されないゾーンとして運用するのが正しいのであれば、以下の点を確認する。

- ゾーンの署名時に、dnssec-signzone コマンドを実行したときに、-A オプション(opt-out ありの運用)をつけずにゾーンファイルが署名されていることが考えられる。

  dnssec-signzone コマンドに –A を指定して実行してゾーンファイルの署名をやり直し、BIND に再読み込みさせる。
- このとき、署名前のゾーンファイルには、署名されない子ゾーン名の DS レコードが設定されていないことを確認する。

2. フルリゾルバ側

確認項目 F-2. DNSSEC 対応フルリゾルバの利用による AD ビットの確認

権威サーバ側が DNSSEC 対応しており(権威を持つゾーンに署名をしている)、フルリゾルバ側も DNSSEC 対応をしている場合、スタブリゾルバからフルリゾルバに DNSSEC を有効にして検索したときに得られたレスポンスに AD ビットが立っていること。

前提事項：

- DNSSEC 対応の権威サーバを構築済み、または利用可能であること。
- DNSSEC 対応のフルリゾルバを構築済みであること。

確認方法：

dig コマンドに +dnssec オプションをつけて問い合わせを行う。このとき、以下を指定する。

- @の横には、DNSSEC 対応のフルリゾルバを指定する。
- 権威サーバが保持している、権威あるゾーンに含まれているホスト名を指定する。
  (下記の例では、www.example.jp としている)

得られたレスポンスのフラグ部に ad が含まれていることを確認する。

```
$ dig +dnssec @192.0.2.1 www.example.jp A

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.1 www.example.jp
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6523
;; flags: qr aa rd ad; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.example.jp.                        IN      A

;; ANSWER SECTION:
www.example.jp.             60      IN      A       192.0.2.2
www.example.jp.             60      IN      RRSIG   A 5 3 60 20091126212001 20091116212001
54842 example.jp. rERzPwquRtfjCa3PbOzTEehAC9HHvtoy8ShpknuvdW4GO/YtGmeiLcNz kDcoKiaaWaAOONiZ
WMv8aDZOG48nkaqehPabdttQm+JNuua8gePXROeF pQl5h8xjlHUPGpq6eCR+bNA+JeirCKkejeVDpjjeAb5h/bZM7d
8bahMs N7Y=

;; AUTHORITY SECTION:
example.jp.                 60      IN      NS      ns.example.jp.
example.jp.                 60      IN      RRSIG   NS 7 2 60 20091218082047 20091118082047
 25277 example.jp. PqtfQNztfJEVaRZrgX353fRtc/WyKaSPAKFYwiqEeUoXoyl+OqPgxsdC dxjJeCW2+HORMVW
jKhVpr+vOrfRa2A==
```

トラブルシューティング：

1. AD ビットが立っていない：

原因 1:

その権威サーバのもつリソースレコードが検証できなかったことを表す。

以下の点を参照し、権威サーバのリソースレコードが適切かどうかを確認する。

確認項目 A-47. DS レコードのアルゴリズムは対応する DNSKEY レコードのものに一致するべき

確認項目 A-76. 子ゾーンが署名付きゾーンの場合、委任点に DS レコードが存在すべき

確認項目 A-78. DS は子ゾーンの頂点にあってはならない

確認項目 A-79. DS レコードは子ゾーン頂点の DNSKEY レコードを参照すべき

確認項目 A-81. 子ゾーンの DS の TTL は、委任 NS の TTL と一致すべき

原因 2:

権威サーバ、フルリゾルバの設定ファイルにおいて、DNSSEC が無効にされていないことを確認する。

以下を参照し、DNSSEC が有効になっていることを確認する。

確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確

認

2. 応答が返ってこない

原因 1:
DNSSEC を有効にした場合の通信が成功していない可能性がある。
+dnssec オプションをはずして問い合わせを行い、応答が返ってくることを確認する。
・応答が返ってくる
以下の 2 点を確認する。

- 途中の経路で EDNS0 による通信が不可能になっている、あるいは TCP/53 番ポートが遮断されていることが考える。以下を確認する。
  確認項目 F-85. セキュリティ対応フルリゾルバは EDNS0 による UDP 通信が可能であること
  確認項目 F-86. セキュリティ対応フルリゾルバは 1220 バイトの UDP メッセージをサポートしていること
  確認項目 F-87. セキュリティ対応フルリゾルバは 4000 バイトの UDP メッセージをサポートすべき

・応答が返ってこない
DNSSEC 以前の問題が発生している可能性がある。
フルリゾルバに対して ping コマンド等を実行し、ネットワークが到達可能か確認する。

確認項目 F-27. RRSIG レコードの有効期間終了フィールドが現在時刻より後であること

RRSIG レコードの RDATA の有効期間終了フィールドが現在時刻より後であること。
また、1970 年 1 月 1 日 0 時 0 分 0 秒(UTC)から経過した秒数について、32 ビット符号なし整数あるいは YYYYMMDDHHmmSS の書式であり、フルリゾルバが検証できること。

確認項目 F-28. RRSIG レコードの有効期間開始フィールドが現在時刻より前であること

RRSIG レコードの RDATA の有効期間終了フィールドが現在時刻より前であること。
また、1970 年 1 月 1 日 0 時 0 分 0 秒(UTC)から経過した秒数について、32 ビット符号なし整数 あるいは YYYYMMDDHHmmSS の書式であり、フルリゾルバが検証できること。

前提事項：

・ DNSSEC 対応の権威サーバを構築済みであること。また DNSSEC 対応の権威サーバが利用可能なこと。
この例では、example.jp というゾーンの権威サーバが利用可能とする。

確認方法：

確認手順書 A-27、A-28 を参照し、以下を確認しておく。

- 親側の権威サーバに登録されている RRSIG レコードの有効期間終了フィールド、有効期間開始フィールドが適切に設定されていること。

次に、フルリゾルバに対し、署名付きゾーンの頂点に対する SOA レコードの問い合わせを行う。dig コマンドに +dnssec オプションをつけて問い合わせを行う。このとき、以下を指定する。

・ @の後ろにはフルリゾルバを指定する。
・ 署名付きゾーン名を指定する。この例では example.jp となる。
・ レコードタイプは DNSKEY を指定する。

```
$ dig +dnssec @192.0.2.201 example.jp SOA

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.201 example.jp SOA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44590
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      SOA

;; ANSWER SECTION:
example.jp.             1080    IN      SOA     ns.example.jp. root.example.jp. 2009120700
360 90 60480 8640
example.jp.             1080    IN      RRSIG   SOA 7 2 1080 20091230034908 20091130034908
48272 example.jp. RiFHpCXvCpgrDbRHXPGiCtpEag7hlgQ3YbB6nFQJLhZPuFEkV8540aK2 txyGeQUr801lt7Gg
Kmfl4OUAPxB5zQ==

;; AUTHORITY SECTION:
example.jp.             1080    IN      NS      ns.example.jp.
example.jp.             1080    IN      RRSIG   NS 7 2 1080 20091230034908 20091130034908 4
8272 example.jp. jnLh/XooDvdGH69Rz9lReebH4ZWU5tRi2V+jHmoTauCo/TiT6pXl3osr mYWDYuOrwwUONrc/a
yDyrcgXzVlHuQ==

;; Query time: 6 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Wed Dec  9 15:26:30 2009
;; MSG SIZE  rcvd: 310
```

以下の点を確認する。

1. 得られたレスポンスのフラグ部に AD ビットが立っていること。
2. 得られたレスポンスの ANSWER セクションに SOA レコードに対する RRSIG レコードが含まれていることを確認する(第 5 カラムが SOA であることを確認する)。
3. その RRSIG の有効期間開始時刻（第 10 カラム）が現在の時刻よりも前であり、有効期間終了時刻（第 9 カラム）が現在の時刻よりも後であること(より現実的には検証作業を行う想定の期間よりも後であること)を確認する。

なお、dig コマンドで表示される有効期間開始時刻、有効期間終了時刻および、ゾーンファイルや dnssec-signzone の-s あるいは -e オプションで指定する有効期間開始時刻、有効期間終了時刻は JST ではなく UTC なので注意する。

トラブルシューティング：

1. 得られたレスポンスに、SOA レコードは含まれているが、RRSIG レコードが含まれていない

原因:
フルリゾルバ側か権威サーバ側のどちらかで、DNSSEC が有効になっていない。
そのため、単に権威サーバの署名付きゾーンの SOA レコードの問い合わせだけを行っており、DNSSEC の検証が行われていない。

- フルリゾルバの確認項目 No.F-229 のトラブルシューティングを参照し、権威サーバ側フルリゾルバ側のどちらとも DNSSEC が有効になっているか確認する。

2. 得られたレスポンスに SOA レコードと RRSIG レコードは含まれているが、AD ビットが立っていない

原因:
フルリゾルバにトラストアンカーが適切に設定されておらず、得られたレコードの検証ができていない。

以下の点を確認する。
フルリゾルバの確認手順書 No.154 を確認し、フルリゾルバにトラストアンカーが設定されて example.jp のゾーンを検証できることを確認する。

3. 以下の実行結果のように、問い合わせが失敗する

```
$ dig +dnssec @192.0.2.201 example.jp SOA

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.201 example.jp SOA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 12440
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      SOA

;; Query time: 4 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Wed Dec  9 15:43:25 2009
;; MSG SIZE  rcvd: 39
```

原因 1:
権威サーバに登録されている RRSIG レコードが適切に設定されておらず、フルリゾルバが検証失敗を返している。
主な原因：

- 有効期間開始時刻と有効期間終了時刻が、ともに現在より未来であるなど、有効期間外である
- 有効期間開始時刻と有効期間終了時刻が間違っている

など

以下の点を確認する。

- 権威サーバ側の署名に問題がある可能性がある。権威サーバ側の設定が確認できる場合には、確認手順書 A-27、A-28 をもう一度確認し、権威サーバに登録されている RRSIG レコードが正しく設定されていること。

原因 2:
権威サーバに登録されている RRSIG レコードが、直接権威サーバに問い合わせて正しく設定されていることを確認しているが、フルリゾルバへの問い合せた場合に失敗する場合は、フルリゾルバに古いキャッシュデータが残っている可能性がある。

以下の点を確認する。

- フルリゾルバを再起動するかキャッシュをクリアし、もう一度フルリゾルバへの問い合わせを行う。

原因 3:
フルリゾルバを稼動しているサーバの時刻設定が間違っているため、権威サーバ側の RRSIG レコードの有効期限開始日時、有効期限終了日時が正しくても、フルリゾルバが検証に失敗している。

以下の点を確認する。

- フルリゾルバを稼動しているサーバの時刻設定を、正しい時刻にする。

原因 4:
フルリゾルバに、誤ったトラストアンカーが設定されているため、フルリゾルバが検証に失敗している。

以下を参照し、フルリゾルバに正しいトラストアンカーを設定する。
確認項目 F-154. セキュリティ対応フルリゾルバは少なくとも 1 つの信頼できる公開鍵または DS を設定に組み込める機能を持たねばならない

確認項目 F-47. DS レコードのアルゴリズムは対応する DNSKEY レコードのものに一致するべき

DS レコードを検索して得られた結果の RDATA のアルゴリズムフィールドは、参照先の DNSKEY レコードを生成時に選択したアルゴリズムに対応した値となっており、フルリゾルバが検証できること。

確認項目 F-49. DS レコードのダイジェストは対応する DNSKEY レコードの鍵のハッシュであるべき

DS レコードを検索して得られた結果の RDATA のダイジェストフィールドは、参照先のゾーンの DNSKEY KSK 鍵をハッシュした文字列と同じものとなっており、フルリゾルバが検証できること。

前提事項：

- DNSSEC 対応のフルリゾルバを構築済みであること。
- 署名つきゾーンの権威サーバと、その親ゾーンの権威サーバがそれぞれ DNSSEC 対応として利用可能なこと。
  署名付きゾーンの DS レコードを、親側の権威サーバに登録済みであること。

ドメインの構成：

- DNSSEC 対応の権威サーバがもつ、署名付きの権威あるゾーン名を sub.example.jp とする。
- 親側のゾーン名を、example.jp とする。
- sub.example.jp は、example.jp の子ゾーンになる。

確認方法：

確認手順書 A-47、A-49 を参照し、以下を確認しておく。

- 親側の権威サーバに登録されている署名つきゾーンの DS レコードのアルゴリズムフィールドは、参照先の署名付きゾーンの DNSKEY レコードのアルゴリズムフィールドと一致すること。
- 親側の権威サーバに登録されている署名つきゾーンの DS レコードのダイジェストフィールドは、参照先の署名付きゾーンの DNSKEY KSK 鍵をハッシュした文

字列と同じものとなっていること。

次に、フルリゾルバに対し、署名付きゾーン(sub.example.jp)の DNSKEY レコードを確認する。dig コマンドに +dnssec オプションをつけて問い合わせを行う。このとき、以下を指定する。

- @の後ろにはフルリゾルバを指定する。
- 署名付きゾーン名を指定する。この例では sub.example.jp となる。
- レコードタイプは DNSKEY を指定する。

```
$ dig +dnssec @192.0.2.201 sub.example.jp DNSKEY

; <<>> DiG 9.6.1-P1 <<>> +dnssec @localhost sub.example.jp DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20538
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      DNSKEY

;; ANSWER SECTION:
sub.example.jp.         1080    IN      DNSKEY  257 3 7 AwEAAcEHNbl/iHv9qRd14VMx9uekrzDBc9z
ieWPqfo1oObx3lhwzLqzC 648pOGyRaElgH1RNWjMDiyk2SJ7ah186KE8=
sub.example.jp.         1080    IN      DNSKEY  256 3 7 AwEAAdwtWnQ5304EMg6wKlTO3qDhrsDj7/S
lWgvwtpNOmrP9JCaAExrm mSlT+zOACCBeZCkVa932CF1l6SrM+a3FAns=
sub.example.jp.         1080    IN      RRSIG   DNSKEY 7 3 1080 20100103095601 200912040956
01 23454 sub.example.jp. oEsqlSC1xsUU59BaZqo4UJP62ayFGpoVOGJ1+eCpedBH671Zx8XJbBs4 cSlW5vgkv
zZ63ZHbRUxtOLeiiHMdNA==
sub.example.jp.         1080    IN      RRSIG   DNSKEY 7 3 1080 20100103095601 200912040956
01 27878 sub.example.jp. hgYs2lULRKOq7x2DCegj+TcaFLThRQpN4EDQrk5LfAymK+bSNt76aYHi lW1gk13nY
46TFplqXvq+K+AGv7G9kQ==

;; Query time: 8 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Tue Dec  8 20:26:17 2009
;; MSG SIZE  rcvd: 431
```

以下の点を確認する。

1. 得られたレスポンスのフラグ部に AD ビットが立っていること。
2. 得られたレスポンスに、署名付きゾーンの DNSKEY レコードと RRSIG レコードが含まれていることを確認する。

トラブルシューティング：

1. 得られたレスポンスに、DNSKEY レコードは含まれているが、RRSIG レコードが含まれていない

原因:
フルリゾルバ側か権威サーバ側のどちらかで、DNSSEC が有効になっていない。
そのため、単に権威サーバの署名付きゾーンの DNSKEY レコードの問い合わせだけを行っており、DNSSEC の検証が行われていない。

- フルリゾルバの確認項目 No.229 のトラブルシューティングを参照し、権威サーバ側とフルリゾルバ側のどちらとも DNSSEC が有効になっているか確認する。

2. 得られたレスポンスに DNSKEY レコードと RRSIG レコードは含まれているが、AD ビットが立っていない

原因:
フルリゾルバにトラストアンカーが適切に設定されておらず、得られたレコードの検証ができていない。

以下の点を確認する。
- フルリゾルバの確認手順書 No.154 を確認し、フルリゾルバにトラストアンカーが設定されて example.jp のゾーンを検証できることを確認する。

3. 以下の実行結果のように、問い合わせが失敗する

```
$ dig +dnssec @192.0.2.201 sub.example.jp DNSKEY

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.201 sub.example.jp DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 43333
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.example.jp.                        IN      DNSKEY

;; Query time: 8 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Tue Dec  8 20:22:56 2009
;; MSG SIZE  rcvd: 43
```

原因 1:
親側の権威サーバに登録されている DS レコードと参照先の署名付きゾーンの DNSKEY レコードが正しく設定されておらず、フルリゾルバが検証失敗を返している。

以下の点を確認する。

- 権威サーバ側の確認手順書 A-47、A-49 をもう一度確認し、親側の権威サーバに登録されている DS レコードと参照先の署名付きゾーンの DNSKEY レコードが正しく設定されていること。

原因 2:
親側の権威サーバに登録されている DS レコードと参照先の署名付きゾーンの DNSKEY レコードが、直接それぞれの権威サーバに問い合わせて正しく設定されていることを確認しているが、フルリゾルバへの問い合せた場合に失敗する場合は、フルリゾルバに古いキャッシュデータが残っている可能性がある。

以下の点を確認する。

- フルリゾルバを再起動するかキャッシュをクリアし、もう一度フルリゾルバへの問い合わせを行う。

原因 3:
親側の権威サーバでの署名が有効期限を過ぎている(権威サーバの RRSIG レコードの有効期限終了日時フィールドが現在時刻を過ぎている)。あるいは、有効期限開始日時や有効期限終了日時が妥当な値でない。これらの理由のためフルリゾルバが検証に失敗している。

以下を参照し、フルリゾルバが権威サーバの署名の有効期間を正しく検証できているか確認する。

確認項目 F-27. RRSIG レコードの有効期間終了フィールドが現在時刻より後であること
確認項目 F-28. RRSIG レコードの有効期間開始フィールドが現在時刻より前であること

原因 4:
フルリゾルバを稼動しているサーバの時刻設定が間違っているため、権威サーバ側の署名の有効期限には問題がなくても、フルリゾルバが検証に失敗している。

以下の点を確認する。

- フルリゾルバを稼動しているサーバの時刻設定を、正しい時刻にする。

原因 5:
フルリゾルバに、誤ったトラストアンカーが設定されているため、フルリゾルバが検証に失敗している。

以下を参照し、フルリゾルバに正しいトラストアンカーを設定する。
確認項目 F-154. セキュリティ対応フルリゾルバは少なくとも 1 つの信頼できる公開鍵または DS を設定に組み込める機能を持たねばならない

確認項目 F-85. セキュリティ対応フルリゾルバは EDNS0 による UDP 通信が可能であること

セキュリティ対応フルリゾルバは、スタブリゾルバーフルリゾルバの間、フルリゾルバー権威サーバの間で EDNS0 による UDP 通信が可能であること。
(問い合わせに対するサーバの応答結果が 512 バイトを超えても、EDNS0 による通信が可能であり、スタブリゾルバが正常に応答を受け取れること。

確認項目 F-86. セキュリティ対応フルリゾルバは 1220 バイトの UDP メッセージをサポートしていること

セキュリティ対応フルリゾルバは、スタブリゾルバーフルリゾルバの間、フルリゾルバー権威サーバの間で 1220 バイトの UDP メッセージをサポートしていること。
(問い合わせに対するサーバの応答結果が 1220 バイトを超えても、EDNS0 による通信が可能であり、スタブリゾルバが正常に応答を受け取れること。

確認項目 F-87. セキュリティ対応フルリゾルバは 4000 バイトの UDP メッセージをサポートすべき

セキュリティ対応フルリゾルバは、スタブリゾルバーフルリゾルバの間、フルリゾルバー権威サーバの間で 4000 バイトの UDP メッセージをサポートすべき。
(問い合わせに対するサーバの応答結果が 4000 バイトを超えても、EDNS0 による通信が可能であり、スタブリゾルバが正常に応答を受け取れること。

前提事項：

・ DNSSEC 対応のフルリゾルバを構築済みであること。また DNSSEC 対応の権威サーバが利用可能なこと。

確認方法 1：

応答結果が 512 バイトを超えるような問い合わせをフルリゾルバに対して行う。

dig コマンドに +dnssec オプションをつけて問い合わせを行う。このとき、以下を指定する。

・ @の横にはフルリゾルバを指定する。
・ 署名付きゾーンのゾーン名を指定する。
(下記の例では、example.jp としている)
・ レコードタイプは DNSKEY を指定する。

```
$ dig +dnssec @192.0.2.201 example.jp DNSKEY

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.201 example.jp DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63155
;; flags: qr rd ra; QUERY: 1, ANSWER: 10, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      DNSKEY

;; ANSWER SECTION:
example.jp.             1067    IN      DNSKEY  257 3 7 AwEAAdGObAi0dHfblc4u/lTk/41LmSQvNDE
GybTsWqr4xZEnPvqnhZio ZBK6Au2doBLJE11p4lXuuqFQH1UPYGd5EPE=
example.jp.             1067    IN      DNSKEY  256 3 7 AwEAAagFpChiBjROKURQfqVgEl87BJS/0x8
vrtal9MSg6uY8F5o56usv O9w99sp0UUdlk4/VRoRCl/PMduZR0t5rnqU=
example.jp.             1067    IN      DNSKEY  256 3 7 AwEAAcMbORS8UoU2uVH3D1Bf09jmlGqL0x1
f6FwV9/H9pcuVrypOg4tR pXDqpemc9uYBtF3OxuHAe/xEis/FsV7LFz8=
example.jp.             1067    IN      DNSKEY  256 3 7 AwEAAcOrMNzTadDMUxqfOoXOzCMETFwFboO
sBjRTwf31wdXCTv5z6nBX 2JkQE83PmdKooPLRNdyNUvs8blaUAXUeEos=
example.jp.             1067    IN      DNSKEY  256 3 7 AwEAAeWVxTfOuNyChnZqOja9tWef/69J4T4
MmkRWEQrO37LggfDiuWL3 eoEb3MxXxFgNqmXXZ26VmQXyflkb1NZCyZs=
example.jp.             1067    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 2105 example.jp. hPAddQ9p8r03/HE8k8NPRkBQQeNAcYOeotc+DrdddjONwoFLX9soOHVe WE23CQoEF4Ruen
MLDKK42MiirgM9ug==
example.jp.             1067    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 14730 example.jp. JlPl1rF+OslBONmg5ZlBP1Qw1+rp/54nHLnfR/e89xvOeXtzRXJp9eLs d1EPOQgENYekd
m8L3SG5mx/a3VwrJg==
example.jp.             1067    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 33692 example.jp. W9q5fxgPOwJ+1ki8K91jttfvCb68eK6cghhMShD2AwmsNVMB6zUoOOy2 RnKmOAAqAnbiC
NBZKWXXl7EOtmOemA==
example.jp.             1067    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 44432 example.jp. AtqfXqsTrUhe2MwqxBmOsQdP1lyLTmMcN+BFH9KKUzKHP5Yzu7T88Aj+ OuzTEKwpCDKBN
SzgkK7RlAdHBjlmXA==
example.jp.             1067    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 60458 example.jp. WsajWVtOAajZZL3ZflrSNEXyzFXc8elalzS7trlGQhf2/d7B56g1bUDV ekobSs+z8bFut
93rDcqgGnZujmJtBg==

;; Query time: 2 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Thu Dec  3 16:53:08 2009
;; MSG SIZE  rcvd: 989
```

以下の点を確認する。

1. example.jp ゾーンに対する DNSKEY レコードと RRSIG レコードが応答に含まれていること。
2. 問い合わせの応答結果の一番最後にある「MSG SIZE rcvd:」という箇所において、応答結果のデータ量が 512 を超えていること(この例では 989)。

3. dig コマンドの応答結果の先頭(コマンドラインでコマンドを入力したすぐ下の行)に、「;; Truncated, retrying in TCP mode.」というメッセージが表示されていないこと。
   これは、フルリゾルバからの応答が 512 バイトを超えても、TCP への切り替わりが起きず、UDP で通信が行われていることを表す。
4. 応答結果の OPT PSEUDOSECTION 部において、udp:4096 となっていること。
   これは、フルリゾルバ側の設定において EDNS0 による UDP 通信を行う場合、4096 バイトのデータ量までサポートされていることを表す。

ここまで確認ができれば、フルリゾルバは、EDNS0 による通信により、512 バイトを超えるデータ量であっても UDP 通信を行えていることになる。

トラブルシューティング１：

1. 問い合わせの応答結果の一番最後にある「MSG SIZE rcvd:」という箇所において、応答結果のデータ量が 512 を超えていない

   原因:
   問い合わせのデータ量が 512 バイトに足りていないので、EDNS0 の動作検証としては適していない。
   以下の点を確認する。
   - 512 バイトを超えるような問い合わせを行う。

2. dig コマンドの応答結果の先頭に「;; Truncated, retrying in TCP mode.」というメッセージが表示されている。
   また、応答結果の OPT PSEUDOSECTION 部において、udp:の箇所が 4096 より小さい数字となっている(下記の例では 512)。

```
$ dig +dnssec @192.0.2.201 example.jp DNSKEY
;; Truncated, retrying in TCP mode.

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.201 example.jp DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5053
;; flags: qr rd ra; QUERY: 1, ANSWER: 10, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 512
;; QUESTION SECTION:
;example.jp.                    IN      DNSKEY

;; ANSWER SECTION:
example.jp.             1080    IN      DNSKEY  257 3 7 AwEAAdGObAi0dHfblc4u/lTk/41LmSQvNDE
GybTsWqr4xZEnPvqnhZio ZBK6Au2doBLJE11p4lXuuqFQH1UPYGd5EPE=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAagFpChiBjROKURQfqVgEl87BJS/0x8
vrtal9MSg6uY8F5o56usv O9w99sp0UUdlk4/VRoRCl/PMduZR0t5rnqU=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcMbORS8UoU2uVH3D1Bf09jmlGqL0x1
f6FwV9/H9pcuVrypOg4tR pXDqpemc9uYBtF3OxuHAe/xEis/FsV7LFz8=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcOrMNzTadDMUxqfOoXOzCMETFwFboO
sBjRTwf31wdXCTv5z6nBX 2JkQE83PmdKooPLRNdyNUvs8blaUAXUeEos=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAeWVxTfOuNyChnZqOja9tWef/69J4T4
MmkRWEQrO37LggfDiuWL3 eoEb3MxXxFgNqmXXZ26VmQXyflkb1NZCyZs=
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 2105 example.jp. hPAddQ9p8r03/HE8k8NPRkBQQeNAcYOeotc+DrdddjONwoFLX9soOHVe WE23CQoEF4Ruen
MLDKK42MiirgM9ug==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 14730 example.jp. JlPl1rF+OslBONmg5ZlBP1Qw1+rp/54nHLnfR/e89xvOeXtzRXJp9eLs d1EPOQgENYekd
m8L3SG5mx/a3VwrJg==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 33692 example.jp. W9q5fxgPOwJ+1ki8K91jttfvCb68eK6cghhMShD2AwmsNVMB6zUoOOy2 RnKmOAAqAnbiC
NBZKWXXl7EOtmOemA==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 44432 example.jp. AtqfXqsTrUhe2MwqxBmOsQdP1lyLTmMcN+BFH9KKUzKHP5Yzu7T88Aj+ OuzTEKwpCDKBN
SzgkK7RlAdHBjlmXA==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 60458 example.jp. WsajWVtOAajZZL3ZflrSNEXyzFXc8elalzS7trlGQhf2/d7B56g1bUDV ekobSs+z8bFut
93rDcqgGnZujmJtBg==

;; Query time: 44 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Thu Dec  3 17:01:13 2009
;; MSG SIZE  rcvd: 989
```

原因:

フルリゾルバ側で、EDNS0 による UDP 通信の最大データ量が制限されている可能性がある(この例では 512 バイトに絞られている)。

この例での問い合わせのデータ量は 989 バイトだが、UDP 通信での最大データ量がこれより小さいため、TCP への切り替わりが起きている。

以下の点を確認する。

- フルリゾルバが BIND の場合：

  named.conf の options ブロックにおいて、以下のような 2 行が設定されていない

か確認する(edns-udp-size 文と max-udp-size 文)。

```
edns-udp-size 512;
max-udp-size 512;
```

この数字を 512 より大きな数字に上げるか、edns-udp-size 文と max-udp-size 文を削除する。

named.conf に edns-udp-size 文と max-udp-size 文を明示的に記述しない場合、4096 が設定される。

- フルリゾルバが Unbound の場合：

unbound.conf の options ブロックにおいて、以下のような行が設定されていないか確認してください(edns-buffer-size 文)。

```
edns-buffer-size: 512;
```

この数字を 512 より大きな数字に上げるか、edns-buffer-size 文を削除する。

unbound.conf に edns-buffer-size 文を明示的に記述しない場合、4096 が設定される。

3. dig コマンドの応答が失敗する。

```
$ dig +dnssec @192.0.2.201 example.jp. DNSKEY
;; Truncated, retrying in TCP mode.
;; Connection to 192.0.2.201#53(192.0.2.201) for example.jp. failed: host unreachable.
$
```

原因:

フルリゾルバ側で、EDNS0 による UDP 通信の最大データ量が制限されている可能性があり、TCP への切り替わりが起きている。

しかしスタブリゾルバとフルリゾルバとの間の TCP での通信がブロックされている可能性がある。

以下の点を確認する。

- スタブリゾルバとフルリゾルバとの間で、TCP、あて先ポートが 53 番の通信がブロックされていないか確認する。ブロックされている場合、開放する必要がある(TCP でも通信できることが必須事項)。

ブロックされているかどうかの確認方法については、以下を参照のこと。

確認項目 共通-1. TCP の通信がブロックされていないことの確認

4. dig コマンドの応答結果の先頭に、DNSKEY レコードは含まれているが、RRSIG レコードが含まれていない。

```
$ dig +dnssec @192.0.2.201 example.jp DNSKEY

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.201 example.jp DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35841
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      DNSKEY

;; ANSWER SECTION:
example.jp.             1080    IN      DNSKEY  257 3 7 AwEAAdGObAiOdHfblc4u/lTk/41LmSQvNDE
GybTsWqr4xZEnPvqnhZio ZBK6Au2doBLJE11p4lXuuqFQH1UPYGd5EPE=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAagFpChiBjROKURQfqVgEl87BJS/Ox8
vrtal9MSg6uY8F5o56usv O9w99spOUUdlk4/VRoRCl/PMduZROt5rnqU=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcMbORS8UoU2uVH3D1Bf09jmlGqLOx1
f6FwV9/H9pcuVrypOg4tR pXDqpemc9uYBtF3OxuHAe/xEis/FsV7LFz8=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcOrMNzTadDMUxqfOoXOzCMETFwFboO
sBjRTwf31wdXCTv5z6nBX 2JkQE83PmdKooPLRNdyNUvs8blaUAXUeEos=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAeWVxTfOuNyChnZqOja9tWef/69J4T4
MmkRWEQrO37LggfDiuWL3 eoEb3MxXxFgNqmXXZ26VmQXyflkb1NZCyZs=

;; Query time: 1 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Thu Dec  3 17:59:18 2009
;; MSG SIZE  rcvd: 459
```

原因

フルリゾルバ側か権威サーバ側のどちらかで、DNSSEC が有効になっていない。

そのため、単に権威サーバの署名付きゾーンの DNSKEY レコードの問い合わせだけを行っており、DNSSEC の検証が行われていない。

この場合、応答結果の OPT PSEUDOSECTION の flags の箇所に do ビットが立っているかどうかを確認する。

- do ビットが立っている:
  権威サーバ側が DNSSEC 対応として正しく設定されていない。
  権威サーバ側の確認する。
- do ビットが立っていない:
  フルリゾルバ側が DNSSEC 対応として正しく設定されていない。
  以下の点を確認する。
  - ✧ フルリゾルバが BIND の場合：
    以下を参照し、DNSSEC が有効になっていることを確認する。
    確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認

確認方法 2.

次に、今回と同じ問い合わせを、+bufsize=512 オプションを付加して行う。
このオプションは、スタブリゾルバ側から、EDNS0 UDP 通信での最大バイトサイズを 512 に制限することをフルリゾルバに通知する。

※ dig コマンドの場合、+bufsize= オプションを付加しない場合はデフォルトで 65535 が適用される。

```
$ dig +dnssec +bufsize=512 @192.0.2.201 example.jp DNSKEY
;; Truncated, retrying in TCP mode.

; <<>> DiG 9.6.1-P1 <<>> +dnssec +bufsize=512 @192.0.2.201 example.jp DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3760
;; flags: qr rd ra; QUERY: 1, ANSWER: 10, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      DNSKEY

;; ANSWER SECTION:
example.jp.             1080    IN      DNSKEY  257 3 7 AwEAAdGObAiOdHfblc4u/lTk/41LmSQvNDE
GybTsWqr4xZEnPvqnhZio ZBK6Au2doBLJE11p4lXuuqFQH1UPYGd5EPE=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAagFpChiBjROKURQfqVgEl87BJS/Ox8
vrtal9MSg6uY8F5o56usv O9w99sp0UUdlk4/VRoRCl/PMduZR0t5rnqU=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcMbORS8UoU2uVH3D1Bf09jmlGqLOx1
f6FwV9/H9pcuVrypOg4tR pXDqpemc9uYBtF3OxuHAe/xEis/FsV7LFz8=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAcOrMNzTadDMUxqfOoXOzCMETFwFboO
sBjRTwf31wdXCTv5z6nBX 2JkQE83PmdKooPLRNdyNUvs8blaUAXUeEos=
example.jp.             1080    IN      DNSKEY  256 3 7 AwEAAeWVxTfOuNyChnZqOja9tWef/69J4T4
MmkRWEQrO37LggfDiuWL3 eoEb3MxXxFgNqmXXZ26VmQXyflkb1NZCyZs=
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 2105 example.jp. hPAddQ9p8rO3/HE8k8NPRkBQQeNAcYOeotc+DrdddjONwoFLX9soOHVe WE23CQoEF4Ruen
MLDKK42MiirgM9ug==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 14730 example.jp. JlPl1rF+OslBONmg5ZlBP1Qw1+rp/54nHLnfR/e89xvOeXtzRXJp9eLs d1EPOQgENYekd
m8L3SG5mx/a3VwrJg==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 33692 example.jp. W9q5fxgPOwJ+1ki8K91jttfvCb68eK6cghhMShD2AwmsNVMB6zUoOOy2 RnKmOAAqAnbiC
NBZKWXXl7EOtmOemA==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 44432 example.jp. AtqfXqsTrUhe2MwqxBmOsQdP1lyLTmMcN+BFH9KKUzKHP5Yzu7T88Aj+ OuzTEKwpCDKBN
SzgkK7RlAdHBjlmXA==
example.jp.             1080    IN      RRSIG   DNSKEY 7 2 1080 20091231051538 200912010515
38 60458 example.jp. WsajWVtOAajZZL3ZflrSNEXyzFXc8elalzS7trlGQhf2/d7B56g1bUDV ekobSs+z8bFut
93rDcqgGnZujmJtBg==

;; Query time: 19 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Thu Dec  3 17:11:19 2009
;; MSG SIZE  rcvd: 989
```

以下の点を確認する。

1. dig コマンドの応答結果の先頭(コマンドラインでコマンドを入力したすぐ下の行)に、「;; Truncated, retrying in TCP mode.」というメッセージが表示されていること。
   これは、問い合わせに対するサーバの応答のデータ量(989)がスタブリゾルバ側から指定した UDP の最大データサイズを超えたため、TCP でリトライして通信が行われたことを表す。
   この例では、TCP でリトライして通信が行われているものの、問い合わせの結果は正常に得られているので、TCP への切り替えが正常に行われれていることになる。

トラブルシューティング 2：

1. dig コマンドの応答が失敗する。
   上記トラブルシューティング 1: の 3. と同様、スタブリゾルバとフルリゾルバとの間で TCP の通信がブロックされている可能性がある。
   上記トラブルシューティング 1: の 3.と同様の確認を行う。

確認方法 3：

応答結果が 1220 バイトを超えるような問い合わせをフルリゾルバに対して行う。

```
$ dig +dnssec @192.0.2.201 example.jp DNSKEY

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.201 example.jp DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14096
;; flags: qr rd ra; QUERY: 1, ANSWER: 30, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      DNSKEY

;; ANSWER SECTION:
example.jp.             1068    IN      DNSKEY  256 3 7 AwEAAddmEC+7iTAxQTU7Mm/Oz9N7vrZx4Ob
54+e1y57yaEtcv+oO3db6 RHyZUE8roL6lGteWjCj8CHn9Vz5Q9FmYJ+c=
example.jp.             1068    IN      DNSKEY  256 3 7 AwEAAdyHv3f8QTiktvjXFGql/13GwNlbtMF
ie6hRl5KRetmZx+SswA/H lOHYyAZ2LHOUv36VRnrBWDghY2qOAroknXM=
～略～
example.jp.             1068    IN      RRSIG   DNSKEY 7 2 1080 20100101020006 200912020200
06 60458 example.jp. Fa+LRGd/MNg831wlO9557MosuRhzs2EK9mxeNEDNyxDHCTQlZHj3aYxt roSh5fZNqHx10
CUkqBZ9VhcMkHOknA==
example.jp.             1068    IN      RRSIG   DNSKEY 7 2 1080 20100101020006 200912020200
06 65309 example.jp. BLALrwtCjpTNUJVZRjdr1HgOzwcxONDL/lowMoxkMkAlxVQjsi3cfCEt ODge1sVcZVTKl
w4hTUmXYLRXR6zwOA==

;; Query time: 10 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Thu Dec  3 17:41:09 2009
;; MSG SIZE  rcvd: 2889
```

以下の点を確認する。

1. example.jp ゾーンに対するDNSKEYレコードとRRSIGレコードが応答に含まれていること。
2. 問い合わせの応答結果の一番最後にある「MSG SIZE rcvd:」という箇所において、応答結果のデータ量が1220を超えていること(この例では2889)。
3. dig コマンドの応答結果の先頭(コマンドラインでコマンドを入力したすぐ下の行)に、「;; Truncated, retrying in TCP mode.」というメッセージが表示されていないこと。

トラブルシューティング３：

1. digコマンドの応答結果の先頭に「;; Truncated, retrying in TCP mode.」というメッセージが表示されている。

   原因:
   フルリゾルバ側で、EDNS0 による UDP 通信の最大データ量が制限されている可能性がある。
   UDP通信での最大データ量が小さいため、TCPへの切り替わりが起きている。
   以下の点を確認する。
   - 上記トラブルシューティング 1. の 2.を参照し、フルリゾルバ側の設定で、EDNS0 による UDP 通信の最大データ量が制限されていないか確認する。
     制限をかける場合であっても、少なくとも1220より大きな値を設定する。

2. digコマンドの応答が失敗する。

   原因:
   フルリゾルバからの応答のデータ量がスタブリゾルバとフルリゾルバの間の通信路のMTUを越えており、パケットのフラグメントが起きている可能性がある。
   フルリゾルバ側の設定において、EDNS0 による UDP 通信の最大データ量を 1220 より大きくしていても、通信路の MTU がこれより小さければパケットのフラグメントが発生する。
   また、通信路において、フラグメントが起きたパケットを落としている可能性がある。
   以下の点を確認する。
   - スタブリゾルバとフルリゾルバの間の通信路の MTU が、1220 を越えているかどうか。

1220 より小さい場合、可能であれば、通信路のネットワーク機器の設定を変更し、MTU の値を大きくする。

- スタブリゾルバとフルリゾルバの間の通信路において、フラグメントしているパケットを落とすような設定がされていないか。
  可能であれば、通信路のネットワーク機器の設定を変更し、パケットがフラグメントしていても通すようにする。

確認方法 4：

応答結果が 4000 バイトを超えるような問い合わせをフルリゾルバに対して行う。

応答結果が 4000 バイトを超えるような問い合わせであっても、正常に応答結果が得られることを確認する。

確認項目 F-130. セキュリティ対応フルリゾルバは元の問い合わせの DO ビットに関わらず、再帰検索においては DO ビットを設定していること

セキュリティ対応フルリゾルバは、スタブリゾルバからの問い合わせに DO ビットが設定されているかにかかわらず、再帰検索を行う際は、DO ビットを設定していること。

前提事項：

・ DNSSEC 対応のフルリゾルバを構築済みであること。また DNSSEC 対応の権威サーバが利用可能なこと。
この例では、example.jp というゾーンの権威サーバが利用可能とする。

確認方法：

まず、フルリゾルバがメモリ上に保持しているキャッシュをクリアする。
キャッシュをクリアするには、以下の方法のいずれかで行う。

- フルリゾルバが BIND の場合：
  1. rndc コマンドを以下のように実行する。

```
$ rndc flushname example.jp
```

注：
rndc コマンドは、BIND に付属している、稼動中の BIND をコントロールするためのコマンドである。rndc コマンドで稼働中の BIND に接続するための設定事項については、各種書籍等を参照されたい。

  2. BIND を再起動する。
- フルリゾルバが Unbound の場合：
  1. unbound-control コマンドを以下のように実行する。

```
$ unbound-control flush_zone example.jp
```

注：
unbound-control コマンドは、Unbound に付属している、稼動中の Unbound をコントロールするためのコマンドである。unbound-control コマンドで稼働中の Unbound に接続するための設定事項については、各種書籍等を参照されたい。

  2. Unbound を再起動する。

次に、フルリゾルバに対し、DO ビットを設定せずに DNSSEC 対応の権威サーバが保持するレコードの問い合わせを行う。dig コマンドに+dnssec オプションを指定せずに問い合わせを行う。このとき以下を指定する。

- @の横には、DNSSEC 対応のフルリゾルバを指定する。
- 権威サーバが保持しているホスト名を指定する。
  (下記の例では、www.example.jp としている)

```
$ dig @192.0.2.201 www.example.jp A

; <<>> DiG 9.6.1-P1 <<>> @192.0.2.201 www.example.jp A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27366
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;www.example.jp.                        IN      A

;; ANSWER SECTION:
www.example.jp.         1080    IN      A       192.0.2.2

;; AUTHORITY SECTION:
example.jp.             1080    IN      NS      ns.example.jp.

;; ADDITIONAL SECTION:
ns.example.jp.          1080    IN      A       192.0.2.1

;; Query time: 4 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Mon Dec  7 19:28:36 2009
;; MSG SIZE  rcvd: 82
```

次に、フルリゾルバが保持しているキャッシュの中身を確認する。
フルリゾルバは、スタブリゾルバからの問い合わせに DO ビットが設定されているかにかかわらず、権威サーバに対しての再帰検索は DO ビットを設定して行う。そこで、フルリゾルバのキャッシュの中身を確認し、先ほど問い合わせた www.example.jp の A レコードだけでなく RRSIG レコード等も存在していれば、確認項目 130 を確認できる。

具体的には以下の方法で行う。

- フルリゾルバが BIND の場合：
  1. rndc コマンドを以下のように実行し、キャッシュの中身をファイルにダンプする。

```
$ rndc dumpdb
```

rndc コマンドに dumpdb を指定して実行すると、フルリゾルバがキャッシュしている内容がファイルに出力される。
ファイルは、BIND の named.conf の options ブロックの directory 文で指定されたディレクトリに、named_dump.db というファイル名で出力される(named.conf の options ブロックにおいて、 dump_file 文で明示的に設定していない場合)。

2. 出力されたファイルの中身を確認する。

```
$ cat named_dump.db
;
; Start view _default
;
;
; Cache dump of view '_default'
;
$DATE 20091207103930
; authauthority
example.jp.             426     IN NS   ns.example.jp.
; authauthority
                        426     RRSIG   NS 7 2 1080 20100106071756 (
                                        20091207071756 48272 example.jp.
                                        Eqx2+H31goShwFfWJuF7OlGEei9bmKDdwGAZ
                                        Qc/uKpitAU6NV/n3MialuY4rLJrAbdiZZsD5
                                        r/vRPEbV/w3/kg== )
; authauthority
ns.example.jp.          426     ¥-AAAA  ;-$NXRRSET
                        426     ¥-AAAA  ;-$NXRRSET
                        426     ¥-AAAA  ;-$NXRRSET
                        426     ¥-AAAA  ;-$NXRRSET
; additional
                        426     A       192.0.2.1
; additional
                        426     RRSIG   A 7 3 1080 20100106071756 (
                                        20091207071756 48272 example.jp.
                                        Hms1kas88K4ViiSVafVCm+CRfJlfatkOlfEv
                                        qVAZgSMfDxwdaYk5UhmlsLQy1+lCScCodEeo
                                        gpJNEp9opM+c4A== )
; authanswer
www.example.jp.         426     A       192.0.2.2
; authanswer
                        426     RRSIG   A 7 3 1080 20100106071756 (
                                        20091207071756 48272 example.jp.
                                        mfJKMLZGT2QtvZEwiqPdS6NzOvK579wetLwT
                                        OW22DMkosRXfHd3ASzU60AUQJdYlJn1BXKjW
                                        OlwKfl/TqptU4Q== )
～略～
```

ファイルから、問い合わせに用いたホスト名(この例では www.example.jp)の箇所を探し、www.example.jp の A レコードだけでなく RRSIG レコードもキャッシュに含まれていることを確認する。

- フルリゾルバが Unbound の場合：

1. unbound-control コマンドを以下のように実行し、キャッシュの中身をファ

イルに出力する。

```
$ unbound-control dump_cache > /tmp/dump_cache.txt
```

unbound-control dump_cache コマンドは、キャッシュの中身を標準出力に出力するため、適当なディレクトリにリダイレクトする。

2. 出力されたファイルの中身を確認する。

```
$ cat /tmp/dump_cache.txt
START_RRSET_CACHE
;rrset 1048 1 1 11 4
example.jp.      1048    IN      NS      ns.example.jp.
example.jp.      1048    IN      RRSIG   NS 7 2 1080 20100106071756 20091207071756 48272 exa
mple.jp. Eqx2+H31goShwFfWJuF70lGEei9bmKDdwGAZQc/uKpitAU6NV/n3MialuY4rLJrAbdiZZsD5r/vRPEbV/w
3/kg== ;{id = 48272}
;rrset 1048 2 2 11 4
example.jp.      1048    IN      DNSKEY  256 3 7 AwEAAbPP6UtH6DXnPRmHaymSYntYMpmcE6FCodYPpug
7qHd60VJG1W+vWH6dt4A42PucEgsY0rX9cbCikU26+ytS8ws= ;{id = 48272 (zsk), size = 512b}
example.jp.      1048    IN      DNSKEY  257 3 7 AwEAAb7wzvpkZJbrxDK1cYUindoFPxqu90lSYwvPMsA
eBOX3D4qq/NhSjmDlidBkeyKcfbFXwwnXB48exHvqYfXbfgU= ;{id = 8895 (ksk), size = 512b}
example.jp.      1048    IN      RRSIG   DNSKEY 7 2 1080 20100106071756 20091207071756 8895
example.jp. m8DO2eZ1UedefBJBflJyDtCcVhMF8vNcrd4+fFtvTbXxcmqPa8GTiq7A8l2TrWkeKni+AHEWtsSeN5h
BS+B3HA== ;{id = 8895}
example.jp.      1048    IN      RRSIG   DNSKEY 7 2 1080 20100106071756 20091207071756 48272
 example.jp. ivtgSdq/0ZkaDrWcphE96nQDQuh1PgmdiYlAkDnSV7doGr86PPJhXxjv5dsays9LhvTDEnyNXvl4Es
Obq2q9fg== ;{id = 48272}
;rrset 1048 1 1 11 4
www.example.jp. 1048    IN      A       192.0.2.2
www.example.jp. 1048    IN      RRSIG   A 7 3 1080 20100106071756 20091207071756 48272 exam
ple.jp. mfJKMLZGT2QtvZEwiqPdS6NzOvK579wetLwTOW22DMkosRXfHd3ASzU6OAUQJdYlJn1BXKjWOlwKfl/Tqpt
U4Q== ;{id = 48272}
;rrset 1048 1 1 11 4
ns.example.jp. 1048     IN      A       192.0.2.1
ns.example.jp. 1048     IN      RRSIG   A 7 3 1080 20100106071756 20091207071756 48272 examp
le.jp. Hms1kas88K4ViiSVafVCm+CRfJlfatkOlfEvqVAZgSMfDxwdaYk5UhmlsLQy1+lCScCodEeogpJNEp9opM+c
4A== ;{id = 48272}
END_RRSET_CACHE
～略～
```

ファイルから、問い合わせに用いたホスト名(この例では www.example.jp)の箇所を探し、www.example.jp の A レコードだけでなく RRSIG レコードもキャッシュに含まれていることを確認する。

トラブルシューティング：

1. 出力されたキャッシュの中身に、RRSIG レコードが含まれていない

原因:
フルリゾルバは DO ビットを設定して再帰問い合わせをしているが、権威サーバが DNSSEC 対応として設定されておらず、DNSSEC に関連するレコードを返却していない。

以下の点を確認する。

- dig コマンドに以下のように +dnssec と+norec オプションをつけて、直接権威サーバに対して問い合わせを行い、権威サーバが DNSSEC 対応をしているかどうかを確認する。
  - @の横には、権威サーバを指定する。
  - フルリゾルバに対して行ったホスト名を指定する。
    (下記の例では、www.example.jp としている)

```
$ dig +dnssec +norec @192.0.2.1 www.example.jp A
```

権威サーバが DNSSEC に関連するレコードを返却していない場合は、権威サーバ側を確認する。

確認項目 F-147. セキュリティ対応フルリゾルバの IP 層は IPv4 か v6 かに関わらず、フラグメントされた UDP パケットを正しく処理できなければならない

セキュリティ対応フルリゾルバの IP 層は、フルリゾルバー権威サーバの間で UDP パケットがフラグメントされることがあっても、正しく処理できること。

前提事項：

- DNSSEC 対応のフルリゾルバを構築済みであること。また DNSSEC 対応の権威サーバが利用可能なこと。
  この例では、example.jp というゾーンの権威サーバが利用可能とする。

確認方法：

まず、フルリゾルバと権威サーバとの間の経路の MTU を確認しておく。
MTU を確認するにはいくつか方法があるが、以下では、例として ping コマンドを用いた方法を説明している。
確認項目 共通-4. ping コマンドによる通信経路の MTU の確認

MTU を確認した結果、ここでは例として 1500 であったとする。

次に、応答結果が MTU の値を超えるような問い合わせをフルリゾルバに対して行う。
問い合わせの例は、以下を参照のこと。
確認項目 A-85. セキュリティ対応権威サーバは EDNS0 による UDP 通信が可能であること

```
$ dig +dnssec @192.0.2.201 example.jp DNSKEY

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.201 example.jp DNSKEY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14096
;; flags: qr rd ra; QUERY: 1, ANSWER: 30, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      DNSKEY

;; ANSWER SECTION:
example.jp.             1068    IN      DNSKEY  256 3 7 AwEAAddmEC+7iTAxQTU7Mm/Oz9N7vrZx4Ob
54+e1y57yaEtcv+oO3db6 RHyZUE8roL6lGteWjCj8CHn9Vz5Q9FmYJ+c=
example.jp.             1068    IN      DNSKEY  256 3 7 AwEAAdyHv3f8QTiktvjXFGql/13GwNlbtMF
ie6hRl5KRetmZx+SswA/H lOHYyAZ2LHOUv36VRnrBWDghY2qOAroknXM=
～略～
example.jp.             1068    IN      RRSIG   DNSKEY 7 2 1080 20100101020006 200912020200
06 60458 example.jp. Fa+LRGd/MNg831wlO9557MosuRhzs2EK9mxeNEDNyxDHCTQlZHj3aYxt roSh5fZNqHx10
CUkqBZ9VhcMkHOknA==
example.jp.             1068    IN      RRSIG   DNSKEY 7 2 1080 20100101020006 200912020200
06 65309 example.jp. BLALrwtCjpTNUJVZRjdr1HgOzwcxONDL/lowMoxkMkAlxVQjsi3cfCEt ODge1sVcZVTKl
w4hTUmXYLRXR6zwOA==

;; Query time: 10 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Thu Dec  3 17:41:09 2009
;; MSG SIZE  rcvd: 2889
```

問い合わせの応答結果の一番最後にある「MSG SIZE rcvd:」という箇所において、応答結果のデータ量が MTU を超えていることを確認する(この例では 2889)。

トラブルシューティング：

1. dig コマンドの応答が失敗する。

原因 1:
権威サーバからの応答のデータ量が権威サーバとフルリゾルバの間の通信路の MTU を越えており、パケットのフラグメントが起きている可能性がある。
また、通信路においてフラグメントが起きたパケットを落としている可能性がある。
以下の点を確認する。

- フルリゾルバと権威サーバの間の通信路において、フラグメントしているパケットを落とすような設定がされていないか。
  可能であれば、通信路のネットワーク機器の設定を変更し、パケットがフラグメントしていても通すようにする。
- 同様の確認を、スタブリゾルバとフルリゾルバの間でも行う。

原因 2:
権威サーバとフルリゾルバの間の通信路のフラグメントについては問題ないが、EDNS0 による通信に問題が発生している。

- フルリゾルバの **確認項目 F-85. セキュリティ対応フルリゾルバは EDNS0 による UDP 通信が可能であること** のトラブルシューティング 1.を参照し、フルリゾルバが EDN0 による通信を行えることを確認する。

確認項目 F-154. セキュリティ対応フルリゾルバは少なくとも 1 つの信頼できる公開鍵または DS を設定に組み込める機能を持たねばならない

セキュリティ対応フルリゾルバは少なくとも1つの信頼できる公開鍵またはDSを設定できることが可能であり、その公開鍵のゾーンを検証できること。

前提事項：

・ DNSSEC 対応のフルリゾルバを構築済みであること。また DNSSEC 対応の権威サーバが利用可能なこと。
この例では、example.jp というゾーンの権威サーバが利用可能とする。

確認方法：

フルリゾルバに信頼できる公開鍵を登録するには、いくつかケースが考えられる。

1. ルートゾーン(“.”)の公開鍵を登録する
2. jp ゾーンなどの tld の公開鍵を登録する
3. example.jp ゾーンなど、任意の署名付きゾーンの公開鍵を登録する

どのケースであっても、設定方法や確認手順に大きな違いはないが、ここでは例として、以下のケースで説明する。

- example.jp というゾーンが署名付きとして利用可能である
- example.jp の公開鍵をフルリゾルバに登録する

手順１：

example.jp の公開鍵を安全な方法で取得する。
ここでは、以下の方法で公開鍵を取得する。

- example.jp ゾーンの管理者が dnssec-signzone コマンドでゾーンファイルを署名した際に、以下のような keyset-(ゾーン名) というファイルが作成されているはずである(この例では keyset-example.jp))。

```
$ cat keyset-example.jp.
$ORIGIN .
example.jp              8640    IN DNSKEY 257 3 7 (
                                        AwEAAb7wzvpkZJbrxDK1cYUindoFPxqu90lS
                                        YwvPMsAeBOX3D4qq/NhSjmDlidBkeyKcfbFX
                                        wwnXB48exHvqYfXbfgU=
                                        ) ; key id = 8895
```

- このファイルを安全な方法で受け取る。

手順 2：

このファイルの内容から、必要な項目をフルリゾルバに登録する。

- フルリゾルバが BIND の場合:
  named.conf に trusted-keys というブロックを作成し、受け取った公開鍵のファイルから以下のように転記する。

```
trusted-keys {
        example.jp 257 3 7 "AwEAAb7wzvpkZJbrxDK1cYUindoFPxqu90lS YwvPMsAeBOX3D4qq/NhS
jmDlidBkeyKcfbFX wwnXB48exHvqYfXbfgU=";
};
```

  - 太字の部分(鍵文字列)は、公開鍵のファイルの（）の内容を 1 行にして転記する。
  - 転記元のファイルでは複数行にわたって記述されているが、これを空白スペースで区切って連結する。
  - 鍵文字列はダブルクォーテーションでくくるようにする。
  - 行末の最後のセミコロン";"を忘れないようにする。
- フルリゾルバが Unbound の場合:
  1. 上記の BIND の named.conf に設定した trusted-keys ブロックと同じ内容のテキストファイルを、unbound.conf が置かれているディレクトリと同じ位置に保存する。
     ファイル名は任意でよいが、ここでは trusted-keys-file とする。
  2. unbound.conf の server:ブロックに、以下の 1 行を追記する。
     trusted-keys-file: trusted-keys-file(保存したファイル名)

手順 3：

フルリゾルバを再起動する。

手順 4：

フルリゾルバに対し、署名付きゾーン(この例では example.jp)の SOA レコードの問い合わせを行う。dig コマンドに +dnssec オプションをつけて問い合わせを行う。このとき、以下を指定する。

- @の後ろにはフルリゾルバを指定する。
- 署名付きゾーン名を指定する。この例では example.jp となる。
- レコードタイプは SOA を指定する。

```
$ dig +dnssec @192.0.2.201 example.jp SOA

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.201 example.jp SOA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61805
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      SOA

;; ANSWER SECTION:
example.jp.             1080    IN      SOA     ns.example.jp. root.example.jp. 2009120700
360 90 60480 8640
example.jp.             1080    IN      RRSIG   SOA 7 2 1080 20100107074519 20091208074519
48272 example.jp. RiFHpCXvCpgrDbRHXPGiCtpEag7hlgO3YbB6nFQJLhZPuFEkV854QaK2 txyGeOUr8Q1lt7Gg
Kmfl4OUAPxB5zQ==

;; AUTHORITY SECTION:
example.jp.             1080    IN      NS      ns.example.jp.
example.jp.             1080    IN      RRSIG   NS 7 2 1080 20100107074519 20091208074519 4
8272 example.jp. jnLh/XooDvdGH69Rz9lReebH4ZWU5tRi2V+jHmoTauCo/TiT6pXl3osr mYWDYuOrwwUONrc/a
yDyrcgXzVlHuQ==

;; Query time: 4 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Wed Dec  9 19:14:30 2009
;; MSG SIZE  rcvd: 310
```

以下の点を確認する。

1. 得られたレスポンスのフラグ部に AD ビットが立っていること。
2. 得られたレスポンスの ANSWER セクションに SOA レコードに対する RRSIG レコードが含まれていること（第５カラムが SOA であること)。

トラブルシューティング：

1. 得られたレスポンスに、SOA レコードは含まれているが、RRSIG レコードが含まれていない

   原因:

フルリゾルバ側か権威サーバ側のどちらかで、DNSSEC が有効になっていない。
そのため、単に権威サーバの署名付きゾーンの SOA レコードの問い合わせだけを行っており、DNSSEC の検証が行われていない。

- 確認項目 F-229 のトラブルシューティングを参照し、権威サーバ側とフルリゾルバ側のどちらも DNSSEC が有効になっているか確認する。

2. 得られたレスポンスに SOA レコードと RRSIG レコードは含まれているが、AD ビットが立っていない

原因:
フルリゾルバにトラストアンカーが設定されていない。

以下の点を確認する。

- フルリゾルバが BIND の場合:
  上記の手順 2: を参照し、named.conf の trusted-keys ブロックにトラストアンカーが設定されているかを確認する。
- フルリゾルバが Unbound の場合:
  上記の手順 2: を参照し、トラストアンカーを記述したファイルが存在するかどうか、また unbound.conf にて、そのファイルを参照しているかを確認する。

3. 以下の実行結果のように、問い合わせが失敗する

```
$ dig +dnssec @192.0.2.201 example.jp SOA

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.201 example.jp SOA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 11375
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      SOA

;; Query time: 6 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Wed Dec  9 20:54:44 2009
;; MSG SIZE  rcvd: 39
```

原因:
フルリゾルバに登録したトラストアンカーと、署名付きゾーン(この例では example.jp)

の DNSKEY レコード(KSK 公開鍵)が一致しておらず、フルリゾルバが検証失敗を返している。

以下の点を確認する。

- 確認項目 A-61 を参照し、署名付きゾーンの権威サーバに対し DNSKEY レコードを問い合わせ、トラストアンカーと一致する DNSKEY レコード(KSK 公開鍵)が登録されていることを確認する。
  登録されていない場合、署名付きゾーンの正しい公開鍵を取得しなおす。

確認項目 F-194. 自分自身で署名され REVOKE bit の立った DNSKEY は Revoke される

REVOKE bit が立っており,自分自身で署名された DNSKEY を受信したリゾルバは、対応する DNSKEY を Revoke すること。

確認項目 F-196. Revoke された DNSKEY は trust anchor として使用されない

Revoke された DNSKEY は trust anchor としては使用できないこと。

確認項目 F-201. タイマー期限が過ぎたら新しい鍵は trust anchor に追加されること

タイマー期限が過ぎたら、次に新しい鍵によって正しく署名された DNSKEY RRSet を見たとき、新しい鍵は trust anchor に追加されること。

確認項目 F-202. タイマー期限が来る前に新しい鍵は trust anchor に追加されていないこと

タイマー期限が来る前に、新しい鍵は trust anchor に追加されていないこと。

注：
本来、上記 4 項目の確認を行うためには、セキュリティの連鎖(セキュリティの島)の開始ゾーンの権威サーバを用意し、そのゾーンの KSK 公開鍵を RFC5011 に従って更新することで確認を行うしかない。これらの準備は負担が大きく、またタイマー期限は通常 1 ヶ月もの長期間にわたる。そのためこれらの確認項目を直接的に検証するよりも、フルリゾルバの実装は問題ないものと仮定して、設定が妥当に行われているかのチェックをもって上記項目の確認を行うものとした。

前提事項：

- DNSSEC 対応かつ RFC5011 を実装したフルリゾルバを構築済みであること。BIND であればバージョンが 9.7.x 以上である必要があるため、ここでは BIND 9.7.0rc1 を使用する。
- DNSSEC 対応かつ RFC5011 を実装した権威サーバが利用可能なこと。BIND であればバージョンが 9.7.x 以上である必要があるため、ここでは BIND 9.7.0rc1 を使用す

る。
この例では、example.jp というゾーンの権威サーバが利用可能とする。

・ 権威サーバには、KSK ゾーン鍵が 2 つ以上登録されていること。

確認方法：

フルリゾルバに信頼できる公開鍵を登録するには、いくつかケースが考えられる。

1. ルートゾーン(“.”)の公開鍵を登録する
2. jp ゾーンなどの tld の公開鍵を登録する
3. example.jp ゾーンなど、任意の署名付きゾーンの公開鍵を登録する

どのケースであっても、設定方法や確認手順に大きな違いはないが、ここでは例として、以下のケースで説明する。

- example.jp というゾーンが署名付きとして利用可能である
- example.jp の公開鍵をフルリゾルバに登録する

手順 1：

example.jp の公開鍵を安全な方法で取得する。
ここでは、以下の方法で公開鍵を取得する。

- example.jp ゾーンの管理者が dnssec-signzone コマンドでゾーンファイルを署名した際に、以下のような keyset-(ゾーン名) というファイルが作成されているはずである(この例では keyset-example.jp))。

```
$ cat keyset-example.jp.
$ORIGIN .
example.jp              8640    IN DNSKEY 257 3 7 (
                                        AwEAAb7wzvpkZJbrxDK1cYUindoFPxqu90lS
                                        YwvPMsAeBOX3D4qq/NhSjmDlidBkeyKcfbFX
                                        wwnXB48exHvqYfXbfgU=
                                        ) ; key id = 8895
                        8640    IN DNSKEY 257 3 7 (
                                        AwEAAdVK3UaQVtCx2B9ICIM2DcJtAm4zllll
                                        QMb7QSy+oeAd+1L84dc9u4e6wt1NBjL5fuhj
                                        5Wq2pVbY1kwguMQstWc=
                                        ) ; key id = 23319
```

- このファイルを安全な方法で受け取る。

手順 2：

このファイルの内容から、必要な項目をフルリゾルバに登録する。

- フルリゾルバが BIND の場合:
  named.conf に managed-keys というブロックを作成し、受け取った公開鍵のファイルから以下のように転記する。

```
managed-keys {
        example.jp initial-key 257 3 7 "AwEAAb7wzvpkZJbrxDK1cYUindoFPxqu9OlS YwvPMsAeB0X3D4qq/NhSjmDlidBkeyKcfbFX wwnXB48exHvqYfXbfgU=";
        example.jp initial-key 257 3 7 "AwEAAdVK3UaQVtCx2B9ICIM2DcJtAm4zllll QMb7QSy+oeAd+1L84dc9u4e6wt1NBjL5fuhj 5Wq2pVbY1kwguMQstWc=";
};
```

  - 太字の部分(鍵文字列)は、公開鍵のファイルの（）の内容を 1 行にして転記する。
  - 転記元のファイルでは複数行にわたって記述されているが、これを空白スペースで区切って連結する。
  - 鍵文字列はダブルクォーテーションでくくるようにする。
  - 行末の最後のセミコロン";"を忘れないようにする。

手順 3：

フルリゾルバを再起動する。

手順 4：

フルリゾルバに対し、署名付きゾーン(この例では example.jp)の SOA レコードの問い合わせを行う。dig コマンドに +dnssec オプションをつけて問い合わせを行う。このとき、以下を指定する。

・ @の後ろにはフルリゾルバを指定する。
・ 署名付きゾーン名を指定する。この例では example.jp となる。
・ レコードタイプは SOA を指定する。

```
$ dig +dnssec @192.0.2.201 example.jp SOA

; <<>> DiG 9.7.0rc1 <<>> +dnssec @192.0.2.201 example.jp SOA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41736
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      SOA

;; ANSWER SECTION:
example.jp.             1080    IN      SOA     ns.example.jp. root.example.jp. 2009120700
360 90 60480 8640
example.jp.             1080    IN      RRSIG   SOA 7 2 1080 20100127034942 20091228034942
48272 example.jp. HzS6Et1DNw1KwoxTdtzj/TB1cfSEhhY95XtogTgVyoWUKFZcq6BRRqlh k486sHuOOTQj7nGr
fgVaZilJ8jlAqA==

;; AUTHORITY SECTION:
example.jp.             1080    IN      NS      ns.example.jp.
example.jp.             1080    IN      RRSIG   NS 7 2 1080 20100127034942 20091228034942 4
8272 example.jp. k6JCkLprkxqmQ2GgFlrtUpbkML89KwhrgCvwU8sw+G36OAQpDzVr58Dl 1dDMo4xV3PQbKlAoE
jUslGi2R5dlDg==

;; Query time: 4 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Mon Dec 28 14:57:46 2009
;; MSG SIZE  rcvd: 310
```

以下の点を確認する。

3. 得られたレスポンスのフラグ部に AD ビットが立っていること。
4. 得られたレスポンスの ANSWER セクションに SOA レコードに対する RRSIG レコードが含まれていること（第５カラムが SOA であること)。

トラブルシューティング：

1. 得られたレスポンスに、SOA レコードは含まれているが、RRSIG レコードが含まれていない

   原因:
   フルリゾルバ側か権威サーバ側のどちらかで、DNSSEC が有効になっていない。
   そのため、単に権威サーバの署名付きゾーンの SOA レコードの問い合わせだけを行っており、DNSSEC の検証が行われていない。

   - 確認項目 F-229 のトラブルシューティングを参照し、権威サーバ側とフルリゾルバ側のどちらも DNSSEC が有効になっているか確認する。

2. 得られたレスポンスに SOA レコードと RRSIG レコードは含まれているが、AD ビットが立っていない

原因 1.:
フルリゾルバにトラストアンカーが設定されていない。

以下の点を確認する。

- フルリゾルバが BIND の場合:
  上記の手順 2: を参照し、named.conf の managed-keys ブロックにトラストアンカーが設定されているかを確認する。

原因 2.:
権威サーバ側で、KSK 公開鍵が 1 つしか登録されていない

以下の点を確認する。

- 権威サーバに、KSK 公開鍵が 2 つ以上登録されていることを確認する。

3. 以下の実行結果のように、問い合わせが失敗する

```
$ dig +dnssec @192.0.2.201 example.jp SOA

; <<>> DiG 9.7.0rc1 <<>> +dnssec @192.0.2.201 example.jp SOA
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 11375
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      SOA

;; Query time: 6 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Mon Dec 28 15:30:15 2009
;; MSG SIZE  rcvd: 39
```

原因:
フルリゾルバに登録したトラストアンカーと、署名付きゾーン(この例では example.jp)の DNSKEY レコード(KSK 公開鍵)が一致しておらず、フルリゾルバが検証失敗を返している。

以下の点を確認する。

- 確認項目 A-61 を参照し、署名付きゾーンの権威サーバに対し DNSKEY レコードを問い合わせ、トラストアンカーと一致する DNSKEY レコード(KSK 公開鍵)が登録されていることを確認する。
  登録されていない場合、署名付きゾーンの正しい公開鍵を取得しなおす。

確認項目 F-229. ゾーンの頂点に NSEC3PARAM レコードが存在すること。

署名したゾーンを持つ権威サーバが該当ゾーンに保持する NSEC3PARAM レコードを、DNSSEC 対応のフルリゾルバが参照できること。

前提事項：

- DNSSEC 対応のフルリゾルバを構築済みであること。また DNSSEC 対応の権威サーバが利用可能なこと。
- 権威サーバ側において、DNSSEC の不在証明が NSEC3 レコード形式で行われていること。

確認方法：

フルリゾルバに対し、署名付きゾーンの NSEC3PARAM レコードの問い合わせを行う。

dig コマンドに +dnssec オプションをつけて問い合わせを行う。このとき、以下を指定する。

- @の横にはフルリゾルバを指定する。
  署名付きゾーンのゾーン名を指定する。
  (下記の例では、example.jp としている)
- レコードタイプは NSEC3PARAM を指定する。

```
$ dig +dnssec @192.0.2.201 example.jp NSEC3PARAM

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.201 example.jp NSEC3PARAM
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55468
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      NSEC3PARAM

;; ANSWER SECTION:
example.jp.             0       IN      NSEC3PARAM 1 0 100 AAAA
example.jp.             0       IN      RRSIG   NSEC3PARAM 7 2 0 20091227053717 20091127053
717 48272 example.jp. HQW/cli29v3rYHOZqvt/kuQTWSWG2Rx+LwloSzRoOAHu7GEfLTQ+adaC HLQF3vJRn+Q3
U4DUa8MauelJuc+b+g==

;; AUTHORITY SECTION:
example.jp.             1080    IN      NS      ns.example.jp.
example.jp.             1080    IN      RRSIG   NS 7 2 1080 20091227053717 20091127053717 4
8272 example.jp. MTdauvE1dMaHZE/litFXRSUZFZU+v78oHEhLKcscecjBKCwei9qsNB6X +By2eDolcwkYPH9PF
pzYaRGTqUyeDA==

;; ADDITIONAL SECTION:
ns.example.jp.          1080    IN      A       192.0.2.1
ns.example.jp.          1080    IN      RRSIG   A 7 3 1080 20091227053717 20091127053717 482
72 example.jp. aJcHslgP6nd78Ym5MuiMNRYttcb/6yloiCtBhT/W+kMJklozhQ1MG6NN sAB9xbrPjnjOFbYkr/Q
A6z49GDTtKw==

;; Query time: 3 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Fri Dec  4 14:52:54 2009
;; MSG SIZE  rcvd: 410
```

得られたレスポンスに、署名付きゾーンの NSEC3PARAM レコードと RRSIG レコードが含まれていることを確認する。

トラブルシューティング：

1. 得られたレスポンスに、NSEC3PARAM レコードは含まれているが、RRSIG レコードが含まれていない

```
$ dig +dnssec @192.0.2.201 example.jp NSEC3PARAM

; <<>> DiG 9.6.1-P1 <<>> +dnssec @192.0.2.201 example.jp NSEC3PARAM
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59051
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;example.jp.                    IN      NSEC3PARAM

;; ANSWER SECTION:
example.jp.             0       IN      NSEC3PARAM 1 0 100 AAAA

;; AUTHORITY SECTION:
example.jp.             1080    IN      NS      ns.example.jp.

;; ADDITIONAL SECTION:
ns.example.jp.          1080    IN      A       192.0.2.1

;; Query time: 3 msec
;; SERVER: 192.0.2.201#53(192.0.2.201)
;; WHEN: Fri Dec  4 15:05:05 2009
;; MSG SIZE  rcvd: 92
```

原因:

フルリゾルバ側か権威サーバ側のどちらかで、DNSSEC が有効になっていない。

そのため、単に権威サーバの署名付きゾーンの NSEC3PARAM レコードの問い合わせだけを行っており、DNSSEC の検証が行われていない。

この場合、応答結果の OPT PSEUDOSECTION の flags の箇所に do ビットが立っているかどうかを確認する。

- do ビットが立っている:
  権威サーバ側が DNSSEC 対応として正しく設定されていない。
  権威サーバ側の確認を行う。
- do ビットが立っていない:
  フルリゾルバ側が DNSSEC 対応として正しく設定されていない。
  以下の点を確認する。
  - フルリゾルバが BIND の場合 :
    以下を参照し、DNSSEC が有効になっていることを確認する。
    確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認

3. 共通項目

確認項目 共通-1. TCP の通信がブロックされていないことの確認

あるホストからあるホストに対し、TCP のあて先ポートが任意である通信がブロックされているかどうかの確認方法について説明する。

前提事項：

通信先のホストで、通信を受け付けるサーバプログラム(DNS サーバなど)が起動していること。

確認方法：

片方のホストにログインし、telnet コマンドを以下のように実行する。
たとえば 192.0.2.1 というホストに対し、TCP のあて先ポートが 53 番の通信について確認するには、以下のように行う。

```
$ telnet 192.0.2.1 53
Trying 192.0.2.1...
Connected to 192.0.2.1 (192.0.2.1).
Escape character is '^]'.
```

このままコマンドプロンプトがしばらく返ってこない状態になれば、通信はブロックされていないことになる。

telnet コマンドの実行結果が、以下のような場合は通信がブロックされているか、通信先のホストでサーバプログラムが起動していない可能性がある。

```
$ telnet 192.0.2.1 53
Trying 192.0.2.1...
telnet: connect to address 192.0.2.1: No route to host
telnet: Unable to connect to remote host: No route to host
$
```

```
$ telnet 192.0.2.1 53
Trying 192.0.2.1...
telnet: connect to address 192.0.2.1: No route to host
telnet: Unable to connect to remote host: No route to host
$
```

通信先のホストや途中の経路で、通信を遮断するような設定がされていないかを確認する。
また、サーバプログラムが起動しているかどうかを確認する。

確認項目 共通-2. BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認

dnssec-signzone コマンドでゾーンファイルを署名した後、BIND の設定において、署名したゾーンファイルが BIND に読み込まれていることの確認方法について説明する。

確認方法：

bind の named.conf のゾーンファイルの指定で、署名したファイルが指定されていることを確認する。

```
zone " example.jp" IN {
        type master;
        file "example.zone.signed";
};
```

file “～” の箇所で、dnssec-signzone コマンドにより生成された署名後のファイルを正しく指定していることを確認する。

ゾーンファイルの署名がまだであれば、dnssec-signzone コマンドを実行してゾーンファイルを署名し、上記のとおり署名後のゾーンファイルを BIND に指定する。

正しく指定しているのであれば、BIND を再起動して、ファイルの内容を BIND に読み込ませる。

確認項目 共通-3. BIND の設定ファイルにおいて DNSSEC が有効になっていることの確認

BIND の設定ファイルにおいて、DNSSEC が有効になっていることの確認方法について説明する。

確認方法：

BIND の設定ファイルの named.conf の options ブロックにおいて、以下の記述がされていないことを確認する。

```
dnssec-enable no;
dnssec-validation no;
```

この 2 行が記述されていない、あるいは yes となっていれば問題ない。

確認項目 共通-4. ping コマンドによる通信経路の MTU の確認

あるホスト間の通信経路の MTU を、ping コマンドを用いて調べる方法を説明する。

確認方法：

手順 1:
調べたい通信経路のうち、片方のサーバにログインする。

手順 2:
通信相手のホストに対し、ping コマンドを以下のように実行する。

```
$ ping -M do -c（回数）-s（パケットサイズ）（通信相手）
```

オプションの簡単な説明：

- -M do ： IP ヘッダにフラグメンテーション不可(DF)を立てる。
- -c (回数)：指定された回数だけパケットを送る
- -s (パケットサイズ)：送るパケットのサイズ

パケットサイズを変えながらこのコマンドを何度か実行することで、経路の MTU を確認することができる。

注：
上記は、今回の検証環境である CentOS release 5.4 で行った場合である。
ping コマンドのオプションは実行する OS によって一部異なるので、環境によっては当該機能がない場合がある。

たとえば以下の実行例では、1472 バイトの ping は成功している。

```
$ ping -M do -c 2 -s 1472 192.0.2.1
PING 10.0.0.2 (192.0.2.1) 1472(1500) bytes of data.
1480 bytes from 192.0.2.1: icmp_seq=1 ttl=64 time=0.255 ms
1480 bytes from 192.0.2.1: icmp_seq=2 ttl=64 time=0.292 ms

--- 192.0.2.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.255/0.273/0.292/0.024 ms
```

- -c オプションで指定した回数と同じ数のパケットを受け取っている
  (2 received)
- パケットロスが 0%となっている
  (0% packet loss)

手順 3:

次に、パケットサイズを変更して ping コマンドを実行する。

たとえば以下の実行例では、1473 バイトの ping に失敗している。

```
$ ping -M do -c 2 -s 1473 192.0.2.1
PING 192.0.2.1 (192.0.2.1) 1473(1501) bytes of data.
From 192.0.2.201 icmp_seq=1 Frag needed and DF set (mtu = 1500)
From 192.0.2.201 icmp_seq=1 Frag needed and DF set (mtu = 1500)

--- 10.0.0.2 ping statistics ---
0 packets transmitted, 0 received, +2 errors
```

- フラグメントが発生しているが、DF ビットがセットされている
  (Flag needed and DF set)
- -c オプションで指定した回数と同じ数のエラーが出ている
  (+2 errors)

今回の例では 1472 バイトの送信には成功しているが 1473 バイトの送信には失敗しているので、MTU は 1500(1472 に IP ヘッダ等 28 を加えた数値)となる。
実際には –s オプションに指定するパケットサイズを変えながら何度かコマンドを実行することで経路の MTU を確認する。

また、ping コマンドの環境によっては MTU の値を表示してくれる場合もある。
(上記実行例の点線の丸の部分)

V.　さいごに

1. 本手順書の扱いについて

本手順書は株式会社日本レジストリサービス（以下 JPRS）が作成したものであり、著作権などの関係権利は JPRS が保有する。

なお、本書の内容確認については以下の組織の協力を得ており、ここに感謝の意を表する。

- Infoblox 株式会社
- KDDI 株式会社
- NEC BIGLOBE 株式会社
- NEC アクセステクニカ株式会社
- エヌ・ティ・ティ・コミュニケーションズ株式会社
- ジュニパーネットワークス株式会社
- ソネットエンタテインメント株式会社
- ソフトバンクテレコム株式会社
- ニフティ株式会社
- ヤマハ株式会社
- 株式会社インターネットイニシアティブ

敬称略 JIS 表記順

Appendix A. DNSSEC機能確認手順書の確認項目一覧

項目の抽出の基準点：

・RFCで MUST とされているものを確認するために、DNSオペレータにとって最低限必要と考えたものを残した。
・RFCで MUST とされているが、ネームサーバの実装内部についての項目は除外した。
・RFCで MUST とされているが、bind のツールや実装を使うことで解決できる(満たされる)ものは除外した。

×除外 ○対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
×	validatorの実装がD0ビットのセットを行わなくするオプションが存在しない	1	RFC4033		クライアントが DNSSEC を有効にしてクエリを要求した場合は、EDNSヘッダ内にDO ビットがたっていること。			
○	サーバ側がdnssec-enableでないかもしれない	2	RFC4033		ネームサーバ側がDNSSEC対応で、リゾルバ側も DNSSEC　対応の場合、リゾルバからサーバに DNSEC を有効にして検索したときに得られたレスポンスに AD ビットがたっていること。	○		権威サーバとフルリゾルバをまとめてひとつにしている
○	通信路やフルリゾルバ設定でEDNS0が落ちるかもしれない	3	RFC4033		EDNS0 による udp 通信が可能であること。	○	○	No.85に統合
○	通信路は1220オクテットのUDPパケットをフラグメントするかもしれない	4	RFC4033		1220オクテットのUDPパケット通信でフラグメントが起きないこと。	○	○	No.86に統合
○	通信路は4000オクテットのUDPパケットを通信できないかもしれない	5	RFC4033		4000オクテットのUDPパケット通信でも通信が可能なこと。	○	○	No.87に統合
×	bindに問題の発生し得るオプションがない	6	RFC4033		DNSSECに対応していない委譲先のゾーンに対する問い合わせを受け付けたとき、正しく返せること。 このとき、ADビットがたっていなこと。	○		
×	bind/dnssec-signzoneに問題の発生し得るオプションがない	7	RFC4034		DNSKEY レコードのタイプ値は 48 であること。	○	○	
×	dnssec-signzone -z あるいは update-check-kskオプションによって正しくならない可能性がある	8	RFC4034		ゾーン鍵(ZSK 公開鍵)であるDNSKEY レコードの RDATA のフラグフィールドの 7ビットは 1 であること。	○	○	
×	dnssec-signzone -z あるいは update-check-kskオプションによって正しくならない可能性がある	9	RFC4034		ゾーン鍵(ZSK 公開鍵)でないDNSKEY レコード(KSK鍵を)の RDATA のフラグフィールドの 7ビットは 0 であること。	○	○	
×	dnssec-signzone -z あるいは update-check-kskオプションによって正しくならない可能性がある	10	RFC4034		DNSKEYレコードの RDATA のフラグフィールドの 15ビットが 1 である場合、その DNSKEY レコードはゾーン鍵であること(7ビットが 1 であること)。	○	○	
×	bindのリゾルバが正常に処理しなくなるオプションが存在しない	11	RFC4034		DNSKEYレコードの RDATA のフラグフィールドの 15ビットが 1 であり、かつ 7ビットが 0 の場合、リゾルバはそのレコードを不正なものとしなければ		○	
×	dnssec-keygenに所定の他のオプションを指定する術がない	12	RFC4034		DNSKEY レコードの RDATA のフラグフィールドの0-6 と 8-14 ビットは、常に0となっていること。	○	○	
	dnssec-keygenはプロトコル番号を明示指定するオプションがある	13	RFC4034		DNSKEY レコードの RDATA のプロトコルフィールドは、常に3であること。	○	○	
×	bindのリゾルバが正常に処理しなくなるオプションが存在しない	14	RFC4034		DNSKEY レコードの RDATA のプロトコルフィールドが3でない場合、リゾルバはそのレコードを不正なものとしなければならない。 (status が NOERROR となっていない？ Ad ビットがたっていない?)		○	
×	鍵作成時のアルゴリズムをオーバーライドするオプションは存在しない	15	RFC4034		DNSKEY レコードの RDATA のアルゴリズムフィールドは、公開鍵作成時に選択したアルゴリズムに対応した値となっていること。	○	○	
×	表示の問題なので技術検証のスコープ外	16	RFC4034		DNSKEY レコードの RDATA のフラグフィールドは符号なし10進数で表記されていること。 (現状では 0, 256, 257)	○	○	

×除外 〇対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
×	挙動をオーバーライドするオプションは存在しない	17	RFC4034		署名つきゾーンにおいて CNAME レコードが存在する場合、その CNAME レコードに対する RRSIG レコードも存在しなければならない。			No.82と類似
×	挙動をオーバーライドするオプションは存在しない	18	RFC4034		RRSIG レコードのタイプ値は 46 であること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	19	RFC4034		RRSIG レコードのクラスは、署名対象のレコードのクラスと同じであること。			
×	挙動をオーバーライドするオプションは存在しない	20	RFC4034		RRSIG レコードの TTL 値は、署名対象のレコードの TTL 値と同じであること。			
×	挙動をオーバーライドするオプションは存在しない	21	RFC4034		RRSIG レコードの署名検証タイプフィールドは、署名対象のレコードのタイプと同じであること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	22	RFC4034		RRSIG レコードの RDATA のアルゴリズムフィールドは、署名生成時に選択したアルゴリズムに対応した値となっていること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	23	RFC4034		対象オーナ名が固有のホストの場合、RRSIG レコードの RDATA のラベルフィールドは 検索時に用いたオーナ名のラベル数であること。 (www.example.jp であれば3 であること)	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	24	RFC4034		対象オーナ名がワイルドカードの場合、RRSIG レコードの RDATA のラベルフィールドは、(検索時に用いたオーナ名 - 1)であること。 (www.example.jp で検索したとき、ワイルドカードであれば 2 であること)	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	25	RFC4034		RRSIG レコードの RDATA のラベルフィールドは、検索時に用いたオーナ名のラベル数以下であること。			
×	挙動をオーバーライドするオプションは存在しない	26	RFC4034		検索時に用いたオーナ名がルート(".")の場合、RRSIG レコードの RDATA のラベルフィールドは 0 であること。 (ルート(".") は含まれない)	〇	〇	
〇	表記の問題ではあるが、値を明示的に指定するオプションが存在するので、運用で誤った値が指定できてしまうため	27	RFC4034		RRSIG レコードの RDATA の有効期間終了フィールドは、1970年1月1日0時0分0秒(UTC)から経過した秒数について、32ビット符号なし整数 あるいは YYYYMMDDHHmmSS の書式であること。	〇	〇	
〇	表記の問題ではあるが、値を明示的に指定するオプションが存在するので、運用で誤った値が指定できてしまうため	28	RFC4034		RRSIG レコードの RDATA の有効期間開始フィールドは、1970年1月1日0時0分0秒(UTC)から経過した秒数について、32ビット符号なし整数 あるいは YYYYMMDDHHmmSS の書式であること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	29	RFC4034		RRSIG レコードの RDATA の署名者名フィールドは、そのゾーン名となっていること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	30	RFC4034		RRSIG レコードの RDATA のオリジナルTTLフィールドは、そのレコードの TTL と同一であること。	〇	〇	
×	表記の問題なので検証作業のスコープ外	31	RFC4034		RRSIG レコードの署名対象フィールドは、ニーモニックで表記されていること。 ニーモニックが不明な場合は RFC3597 のセクション5. で規定する表記が使用されていること。	〇	〇	
×	表記の問題なので検証作業のスコープ外	32	RFC4034		RRSIG レコードのアルゴリズムフィールド値は、RFC4034 付録A.1 で指定されているニーモニックのどれか、あるいは符号無し10進整数となっていること。	〇	〇	
×	表記の問題なので検証作業のスコープ外	33	RFC4034		RRSIG レコードのオリジナルTTLフィールドは、符号無し10進整数となっていること。	〇	〇	
×	表記の問題なので検証作業のスコープ	34	RFC4034		RRSIG レコードの鍵タグフィールドは、符号無し10進整数であること。	〇	〇	
×	表記の問題なので検証作業のスコープ	35	RFC4034		RRSIG レコードの署名者名フィールドは、ドメイン名で表記されていること。	〇	〇	
×	表記の問題なので検証作業のスコープ外	36	RFC4034		RRSIG レコードの署名フィールドは、Base64で符号化されたテキストであること。	〇	〇	
×	表記の問題なので検証作業のスコープ外	37	RFC4034		RRSIG レコードの署名フィールドに空白文字が含まれていてもエラーとならないこと。		〇	
×	挙動をオーバーライドするオプションは存在しない	38	RFC4034	NSEC	NSEC レコードのタイプ値は 47 であること。	〇	〇	

×除外 ○対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
×	挙動をオーバーライドするオプションは存在しない	39	RFC4034	NSEC	NSEC レコードの TTL フィールドは、そのゾーンの SOA レコードの最小TTLフィールドと同じ値を持っていること。	○	○	
×	挙動をオーバーライドするオプションは存在しない	40	RFC4034	NSEC	NSEC レコードの次ドメイン名フィールドは、そのゾーンが持つドメイン名を正規的な順序に並び替えた際に、検索に用いたオーナ名の次の名前とな	○	○	
×	挙動をオーバーライドするオプションは存在しない	41	RFC4034	NSEC	ゾーン内にある最後のオーナ名のNSEC レコードの次ドメイン名フィールドは、SOA レコードのオーナ名となること。	○	○	
×	挙動をオーバーライドするオプションは存在しない	42	RFC4034	NSEC	NSEC レコードの次ドメイン名の並び順は、RFC4034 Section 6.1 のとおりとなっていること。	○	○	
×	挙動をオーバーライドするオプションは存在しない	43	RFC4034	NSEC	レコードのオーナ名が(グルーレコードのように)ゾーンで権威を持たない場合、ゾーン内にそのオーナ名で権威を持つレコードがひとつも存在しなければ、NSEC レコードの次ドメイン名として使われていないこと。	○	○	
×	挙動をオーバーライドするオプションは存在しない	44	RFC4034	NSEC	あるオーナ名のNSEC レコードのタイプビットマップフィールドには、そのオーナ名のレコードが持つタイプがすべて含まれていること。	○	○	
×	挙動をオーバーライドするオプションは存在しない	45	RFC4034	NSEC	あるオーナ名のNSEC レコードのタイプビットマップフィールドには、そのオーナ名のレコードが持たないタイプは含まれていないこと。	○	○	
×	挙動をオーバーライドするオプションは存在しない	46	RFC4034		DS レコードのタイプ値は 43 であること。	○	○	
○	親ゾーンにDSを登録するために手作業での編集を行う可能性がありうる	47	RFC4034		DS レコードの RDATA のアルゴリズムフィールドは、参照先のDNSKEY レコードを生成時に選択したアルゴリズムに 対応した値となっていること。	○	○	
×	dnssec-dsfromkeyコマンド等で設定可能になる。また親ゾーンにDSを登録するために手作業での編集を行う可能性があり	48	RFC4034		DS レコードの RDATA のダイジェストタイプフィールドは、参照先のDNSKEYレコードを生成時に選択したダイジェストアルゴリズムに対応した値となっていること。	○	○	
○	親ゾーンにDSを登録するために手作業での編集を行う可能性がありうる	49	RFC4034		DS レコードの RDATA のダイジェストフィールドは、参照先のゾーンのDNSKEY KSK 鍵をハッシュした文字列と同じものとなっていること。	○	○	
○	親ゾーンにDSを登録するために手作業での編集を行う可能性がありうる	50	RFC4034		DS レコードの参照する DNSKEY レコードは、DNSSEC ゾーン鍵でなければならない(参照先の DNSKEY レコードの RDATA のフラグフィールドの 7 ビット目に 1 がたっていること)。	○		
×	挙動をオーバーライドするオプションは存在しない	51	RFC4034		DS レコードの参照する DNSKEY レコードを検索した結果、DNSSEC ゾーン鍵でなかった場合、検証が不成立となること。		○	
×	表記の問題なので検証作業のスコープ	52	RFC4034		DS レコードの鍵タグフィールドは、符号無し10進整数であること。	○	○	
×	表記の問題なので検証作業のスコープ外	53	RFC4034		DS レコードのアルゴリズムフィールド値は、RFC4034 付録A.1 で指定されているニーモニックのどれか、あるいは符号無し10進整数となっていること。	○	○	
×	表記の問題なので検証作業のスコープ	54	RFC4034		DS レコードのダイジェストフィールドは、符号無し10進整数であること。	○	○	
×	表記の問題なので検証作業のスコープ外	55	RFC4034		DS レコードのダイジェストは大文字小文字を区別しない、16進数の並びで表記されていること。	○	○	
×	挙動をオーバーライドするオプションは存在しない	56	RFC4034		DNSサーバに、重複したRRsetが登録されている場合、プロトコルエラーとして処理されること。	○	○	
×	bindは全ての必須アルゴリズムを実装しているので問題ない	57	RFC4034		DNSSEC対応ネームサーバは、必須となるアルゴリズムをすべて実装していること。	○	○	
○	ゾーンファイルの設定によりDNSKEY RRが含まれない可能性がある	58	RFC4035		署名に使用した鍵がゾーン頂点のDNSKEY RRに含まれているべき。	○		2.1 ゾーンへのDNSKEY RRの付加
×	dnssec-keygenにオーバーライドするオプションがある	59	RFC4035		ゾーン鍵のDNSKEY RRは、RDATAフィールドのゾーン鍵フラグを設定しなければならない。	○		
×	挙動をオーバーライドするオプションは存在しない	60	RFC4035		ゾーン鍵フラグの立っていないDNSKEYをRRSIGの検証に使用してはならない。		○	

×除外 ○対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
○	ゾーンファイルの設定によりSEPである DNSKEY RRが含まれない可能性がある	61	RFC4035		セキュリティの島でない署名付きゾーンにはSEPであるDNSKEYが最低1つ頂点になければならない。	○		
×	挙動をオーバーライドするオプションは存在しない	62	RFC4035		署名付きゾーン内の権威をもつRRSetそれぞれに対し、RRSIGが最低1つなければならない。一致条件は、所有者名・クラス・署名対象タイプ・オリジナルTTL・TTL・ラベル数・署名者名(=ゾーン名)であり、RRSIGのアルゴリズム・署名者名・鍵タグがゾーン鍵を特定。	○		2.2 ゾーンへのRRSIG RRの付加
×	挙動をオーバーライドするオプションは存在しない	63	RFC4035		RRSIGに対するRRSIGは存在してはならない。	○		
×	挙動をオーバーライドするオプションは存在しない	64	RFC4035		署名付きゾーン頂点のNSは署名つきでなければならない。	○		
×	挙動をオーバーライドするオプションは存在しない	65	RFC4035		委任点のNSは署名されてはならない。	○		
×	挙動をオーバーライドするオプションは存在しない	66	RFC4035		委任点のグルーレコードは署名されてはならない。	○		
×	挙動をオーバーライドするオプションは存在しない	67	RFC4035		署名付きゾーンの各RRSetに対し、頂点にあるDNSKEYのうち少なくとも1つを使ったRRSIGがなければならない。	○		
○	親と子のゾーン設定により異る可能性がある	68	RFC4035		署名付きゾーン頂点のDNSKEY自身は親側の委任点にあるDSが示すアルゴリズムによって署名されていなければならない。	○		
×	挙動をオーバーライドするオプションは存在しない	69	RFC4035	NSEC	署名付きゾーンの権威を持つデータおよび委任点のNSを持つ所有者名に対しNSECがなければならない。	○		2.3. ゾーンへのNSEC RRの付加
×	挙動をオーバーライドするオプションは存在しない	70	RFC4035	NSEC	NSEC RRのTTLはSOAのminimumと同じであるべき。	○		
×	挙動をオーバーライドするオプションは存在しない	71	RFC4035	NSEC	NSECとRRSIGしか存在しない所有者名があってはならない。	○		
×	挙動をオーバーライドするオプションは存在しない	72	RFC4035	NSEC	署名前にRRSetが存在しなかった名前に対しNSEC,RRSIGを生成してはならない。	○		
×	挙動をオーバーライドするオプションは存在しない	73	RFC4035	NSEC	NSECのタイプビットマップにはNSECとRRSIGが含まれていなければならない。	○		
×	挙動をオーバーライドするオプションは存在しない	74	RFC4035	NSEC	委任点におけるNSECのタイプビットマップにはNSと権威を持つRRSetのタイプが設定されなければならない。(DS)。	○		
×	挙動をオーバーライドするオプションは存在しない	75	RFC4035	NSEC	委任点におけるNSECのタイプビットマップにはNSと権威を持つRRSetのタイプ以外はクリアされていなければならない。	○		
○	親と子のゾーン設定により異る可能性がある	76	RFC4035		子ゾーンが署名付きゾーンの場合委任点にDS RRSetが存在すべき。	○		2.4. ゾーンへのDS RR付加
×	挙動をオーバーライドするオプションは存在しない	77	RFC4035		署名付きゾーン内のすべてのDSは署名つきでなければならない。	○		
○	ゾーンファイルの設定による存在する可能性がある	78	RFC4035		DSはゾーン頂点にあってはならない。	○		
○	親と子のゾーン設定により異る可能性がある	79	RFC4035		DSは子ゾーン頂点のDNSKEYを参照すべき。	○		
○	親と子のゾーン設定により異る可能性がある	80	RFC4035		署名付きの子ゾーン頂点にあるDNSKEY RRSetは(DSと)対応する秘密鍵で署名されるべき。	○		
○	親と子のゾーン設定により異る可能性がある	81	RFC4035		DS RRSetのTTLは委任NSと一致すべき。	○		

×除外 〇対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
×	挙動をオーバーライドするオプションは存在しない	82	RFC4035	NSEC	署名付きゾーンではCNAMEに対しても適切なRRSIG/NSECが要求される。	〇		
×	挙動をオーバーライドするオプションは存在しない	83	RFC4035	NSEC	RRSIG/NSEC、安全な動的更新をおこなうKEY以外のタイプがCNAMEを持つ所有者名に存在してはならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	84	RFC4035	NSEC	署名付きゾーン内の委任点には所有者名に対するNSEC(およびそれに対するRRSIG)が要求される。	〇		
〇	EDNSをサポートしないオプションがある	85	RFC4035		セキュリティ対応ネームサーバはEDNS0をサポートしなければならない。	〇	〇	3. サーバ側の処理
〇	EDNSをサポートしないオプションがある UDPサイズを指定するオプションがある	86	RFC4035		セキュリティ対応ネームサーバは少なくとも1220バイトのメッセージをサポートしなければならない。	〇	〇	
〇	EDNSをサポートしないオプションがある UDPサイズを指定するオプションがある	87	RFC4035		セキュリティ対応ネームサーバは4000バイトのメッセージをサポートすべき。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	88	RFC4035		セキュリティ対応ネームサーバはOPT RRがない、またはDO=0の問い合わせに対してはRRSIG/DNSKEY/NSECに対しては通常のRRSetのように扱わなければならない(DSを除く)。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	89	RFC4035		セキュリティ対応ネームサーバはDSに対しては上記条件でも特別な処理を行う。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	90	RFC4035		セキュリティ対応ネームサーバは上記条件において、付加的な処理をしてはならない。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	91	RFC4036		セキュリティ対応ネームサーバは親と子の両ゾーンを保持する場合、委任点にあるNSEC/RRSIGに対する問い合わせに対して、一貫性のある応答をしなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	92	RFC4035		セキュリティ対応ネームサーバはDNAMEから合成されたCNAMEに対して署名を生成すべきではない。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	93	RFC4035		セキュリティ対応ネームサーバは問い合わせのCDビットを応答にコピーしなければならない。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	94	RFC4035		セキュリティ対応ネームサーバは問い合わせのADビットを無視しなければならない。	〇	〇	
〇	dnssec-enable でない場合、あるいは ゾーンが署名されていない場合はRRSIGを送信しないので確認する	95	RFC4035		セキュリティ対応権威ネームサーバはDO=1の問い合わせに応答する場合RRSIGを送信すべき。	〇		3.1.1. 応答へのRRSIG RR付加
×	挙動をオーバーライドするオプションは存在しない	96	RFC4035		応答の回答部に署名つきRRSetを付加する場合、ネームサーバはRRSIGも回答部に付加しなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	97	RFC4035		上記の場合、ほかに付加されるRRSetよりもRRSIGのほうが優先度が高く、RRSIGが容量的に付加できない場合、TCビットを設定しなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	98	RFC4035		権威部に署名つきRRSetを付加する場合、対応するRRSIGも権威部に付加しなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	99	RFC4035		上記の場合、ほかに付加されるRRSetよりもRRSIGのほうが優先度が高く、RRSIGが容量的に付加できない場合、TCビットを設定しなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	100	RFC4035		付加情報部に署名つきRRSetを付加する場合、対応するRRSIGも付加情報部に付加しなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	101	RFC4035		付加情報部に必要なRRSIGが入らなかったというだけの理由でTCビットが設定されてはならない。(付加情報部のRRSIGのみ切り捨ててもよい)	〇		

×除外 〇対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
×	挙動をオーバーライドするオプションは存在しない	102	RFC4035		ゾーン頂点のSOAまたはNSに対するDO=1の問い合わせに対し、DNSKEYとそのRRSIGを付加情報部に付加して応答できるが、容量が足りない時はDNSKEYとそのRRSIGの両方を除外しなければならない。	〇		3.1.2. 応答へのDNSKEY RR付加
×	挙動をオーバーライドするオプションは存在しない	103	RFC4035		上記の理由だけでTCビットを設定してはならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	104	RFC4035	NSEC	セキュリティ対応権威ネームサーバは署名つきゾーンに対するDO=1の問い合わせに対し、データ無し・名前エラー・ワイルドカードによる応答・ワイルドカードによるデータ無し場合にNSECを付加しなければならない。	〇		3.1.3. 応答へのNSEC RR付加
×	挙動をオーバーライドするオプションは存在しない	105	RFC4035	NSEC	″データ無し″の場合、SNAME,SCLASSに一致するNSECとそのRRSIGを権威部に付加しなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	106	RFC4035	NSEC	上記付加レコードが容量的に入らない場合TCビットを設定しなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	107	RFC4035	NSEC	″名前エラー″の場合、SNAME,SCLASSをカバーするNSEC、ワイルドカード展開によりSNAME,SCLASSにマッチするRRSetが存在しないことを表すNSEC、およびこれらNSECのRRSIGが権威部に付加されなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	108	RFC4035	NSEC	単一のNSECが完全一致するRRSetとワイルドカード展開によって一致するRRSetの両方が存在しないことを示す場合、付加するNSECとRRSIGはそれぞれ1つずつ権威部に付加するべき。	〇		
×	挙動をオーバーライドするオプションは存在しない	109	RFC4035	NSEC	上記付加レコードが容量的に入らない場合TCビットを設定しなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	110	RFC4035	NSEC	″ワイルドカードによる応答″をする場合、ワイルドカード展開後の回答と、ワイルドカードのRRSIGを回答部に付加しなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	111	RFC4035	NSEC	上記の場合、権威部にSNAME,SCLASSによりよく一致するRRSetがないことを証明するNSECとそのRRSIGが付加されなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	112	RFC4035	NSEC	上記付加レコードが容量的に入らない場合TCビットを設定しなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	113	RFC4035	NSEC	″ワイルドカードによるデータ無し″の場合、SNAME,SCLASSに一致するワイルドカード所有者名をもつRRSetでSTYPEに一致するものはないことを証明するNSECと、SNAME,SCLASSによりよく一致するRRSetがないことを示すNSEC、およびこれらのNSECのRRSIGが権威部に付加されなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	114	RFC4035	NSEC	上記付加レコードが容量的に入らない場合TCビットを設定しなければならない。	〇		
〇	ゾーンにDSを登録する段階で委任点にDSがないミスが発生する可能性がある	115	RFC4035		DO=1の適切な問い合わせに対し、署名つきゾーンのセキュリティ対応権威ネームサーバが参照を返す場合、委任点にDSが存在するときはNSに加えDSとそのRRSIGをともに権威部に付加しなければならない。	〇		3.1.4. 応答へのDS RR付加
〇	ゾーンにDSを登録する段階で委任点にDSがあるミスが発生する可能性がある	116	RFC4035		DO=1の適切な問い合わせに対し、署名つきゾーンのセキュリティ対応権威ネームサーバが参照を返す場合、委任点にDSが存在しないときは、DSがないことを証明するNSECとそのRRSIGを共に返さなければならない。すなわち、NSとNSECとNSECのRRSIGを付加しなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	117	RFC4035		容量的にDS,NSEC,および対応するRRSIGが付加できない場合、TCビットを設定しなければならない。	〇		
×	挙動をオーバーライドするオプションは存在しない	118	RFC4035		ネームサーバはゾーンカットにあるDSへの問い合わせを受信し、子ゾーンに権威を持ち、親ゾーンに権威を持たず、再起検索を行わない場合、権威を持つデータなし応答を返さなければならない。	〇		3.1.5. タイプAXFRまたはIXFRへの問合わせに対する応答
×	挙動をオーバーライドするオプションは存在しない	119	RFC4035		ゾーン転送に対し、レコードを選択的に拒否したり受容したりしてはならない。(転送を許す場合は全体を転送しなければならない)	〇		
×	挙動をオーバーライドするオプションは存在しない	120	RFC4035		ゾーン転送においては、権威を持つDSを付加しなければならない。	〇		

×除外 ○対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
×	挙動をオーバーライドするオプションは存在しない	121	RFC4035	NSEC	権威を持つNSECはゾーン転送に含まれなければならない。	○		
×	挙動をオーバーライドするオプションは存在しない	122	RFC4035	NSEC	親ゾーン転送時には委任点のNSECは親ゾーン含まれなければならない。	○		
×	挙動をオーバーライドするオプションは存在しない	123	RFC4035	NSEC	子ゾーン転送時には頂点のNSECは子ゾーン転送時に含まれなければならない。	○		
×	挙動をオーバーライドするオプションは存在しない	124	RFC4035		同様に権威を持つRRSIGはゾーン転送に含まれなければならない。	○		
×	挙動をオーバーライドするオプションは存在しない	125	RFC4035		セキュリティ対応ネームサーバは権威を持つ応答を生成する際、CDビットをクリアすべき。	○		3.1.6. 権威を持つ応答におけるADビットとCDビット
×	挙動をオーバーライドするオプションは存在しない	126	RFC4035		応答の回答部・権威部に入れられたRRSetがすべて信頼できない限り、セキュリティ対応ネームサーバはADビットを設定してはならない。	○		
×	挙動をオーバーライドするオプションは存在しない	127	RFC4035		ネームサーバが権威を持つゾーンを安全なゾーン転送などの安全な手段で入手できない限り付加的検証なしの信頼をしてはならない。	○		
×	挙動をオーバーライドするオプションは存在しない	128	RFC4035		明示的に設定されていない限り、付加的検証なしの信頼をしてはならない。	○		
×	挙動をオーバーライドするオプションは存在しない	129	RFC4035		再帰検索をサポートするセキュリティ対応ネームサーバは再帰検索によって得たデータを含む応答を生成する際に、CDビットとADビットに関して以下の規則に従わなければならない。		○	
○	再帰ネームサーバのdnssec-enableが有効に設定されていることを確認する必要がある	130	RFC4035		再帰ネームサーバは元の問い合わせのDOビットにかかわらず、再帰検索においてはDOビットを設定しなければならない。		○	3.2. 再帰ネームサーバ
×	挙動をオーバーライドするオプションは存在しない	131	RFC4035		元の問い合わせがDO=0の場合、認証用DNSSEC RRを取り除かなければならない。		○	
×	挙動をオーバーライドするオプションは存在しない	132	RFC4035		ただし、明示的にDNSSEC RRタイプが要求されている場合はこれを取り除いてはならない。		○	
×	挙動をオーバーライドするオプションは存在しない	133	RFC4035		再帰ネームサーバのネームサーバサイドではCDビットを問い合わせから応答にコピーしなければならない。		○	
×	挙動をオーバーライドするオプションは存在しない	134	RFC4035		ネームサーバーサイドで受け取ったCDビットをリゾルバサイドに渡さなければならない。		○	
×	挙動をオーバーライドするオプションは存在しない	135	RFC4035		CDビットがセットされている場合、ローカルな認証ポリシーがきっかけとなる問い合わせを生成したリゾルバが要求するレコードの提供を拒否しているとしても、可能な限りそのデータを返すべき。		○	
×	挙動をオーバーライドするオプションは存在しない	136	RFC4035		リゾルバサイドの不良キャッシュに一致する問い合わせをネームサーバサイドが受信した場合、CDビットが設定されているときは不良キャッシュからデータをかえすべき。		○	
×	挙動をオーバーライドするオプションは存在しない	137	RFC4035		リゾルバサイドの不良キャッシュに一致する問い合わせをネームサーバサイドが受信した場合、CDビットが設定されていない場合はネームサーバはRCODE=2(servfail)を返さなければならない。		○	
×	挙動をオーバーライドするオプションは存在しない	138	RFC4035		リゾルバサイドの不良キャッシュに一致する問い合わせをネームサーバサイドが受信した場合、CDビットが設定されている場合は不良キャッシュを返すべき。		○	
×	挙動をオーバーライドするオプションは存在しない	139	RFC4035		再帰ネームサーバのネームサーバサイドは応答の回答部・権威部に入れられたRRSetがすべて信頼できない限り、セキュリティ対応ネームサーバはADビットを設定してはならない。		○	
×	挙動をオーバーライドするオプションは存在しない	140	RFC4035		ネームサーバサイドは回答部の全RRSetと権威部の適切な否定応答を示すRRが信頼できる場合にだけADビットを設定すべき。		○	
×	挙動をオーバーライドするオプションは存在しない	141	RFC4035		リゾルバサイドは信頼できるかどうかを署名検証の手続きに従って判断しなければならない。		○	

×除外 〇対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
×	挙動をオーバーライドするオプションは存在しない	142	RFC4035		ネームサーバサイドは信頼できるDNAMEから合成されたことがあきらかなCNAMEがRFC2672の規則にしたがって応答に含まれる場合はADビットを設定してよい。		〇	
×	挙動をオーバーライドするオプションは存在しない	143	RFC4035		セキュリティ対応リゾルバはDO=1のOPTレコードを問い合わせに付加しなければならない。		〇	4. リゾルバ側処理
×	挙動をオーバーライドするオプションは存在しない	144	RFC4035		セキュリティ対応リゾルバは最低1220オクテットのメッセージサイズをサポートしなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	145	RFC4035		セキュリティ対応リゾルバは4000オクテットのメッセージサイズをサポートすべき。		〇	
×	挙動をオーバーライドするオプションは存在しない	146	RFC4035		セキュリティ対応リゾルバはOPTのsender's UDP payload sizeで受信可能なメッセージサイズを広報しなければならない。		〇	
〇	実装に係らずOSおよびネットワーク機器によってフラグメントを正しく処理できないことがある	147	RFC4035		セキュリティ対応リゾルバのIP層はIPv4/6かにかかわらずフラグメントされたUDPパケットを正しく処理できなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	148	RFC4035		セキュリティ対応リゾルバは署名検証の仕組みをサポートしなければならない。		〇	4.2. 署名検証のサポート
×	挙動をオーバーライドするオプションは存在しない	149	RFC4035		セキュリティ対応リゾルバは受信した応答すべてに署名検証の仕組みを適用すべき（再帰ネームサーバがCD=1の再帰問い合わせを受け取った場合、その他の理由で検証しないように設定・指示されている場合）。		〇	
×	挙動をオーバーライドするオプションは存在しない	150	RFC4035		セキュリティ対応リゾルバの署名検証サポートにはワイルドカード所有者名の検証が含まれてなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	151	RFC4035	NSEC	ゾーンカットの親側にあるNSECが不足していてそれを取得しようと試みる場合、セキュリティ対応反復型リゾルバは親ゾーンに問い合わせをしなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	152	RFC4035		DSが不足していてそれを取得しようと試みる場合、セキュリティ対応反復型リゾルバは親ゾーンに問い合わせをしなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	153	RFC4035		セキュリティ対応リゾルバは、特定のRRsetが署名つきであると期待すべきかどうかを判定できなければならない。すなわち安全/安全でない/不適正/不確定を識別できなければならない。		〇	
〇	トラストアンカーに設定を失敗している可能性がある	154	RFC4035		セキュリティ対応リゾルバは少なくとも1つの信頼できる公開鍵またはDSを設定に組み込める機能を持たねばならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	155	RFC4035		セキュリティ対応リゾルバは複数の公開鍵またはDSを設定に組み込めるようにすべき。		〇	
×	挙動をオーバーライドするオプションは存在しない	156	RFC4035		セキュリティ対応リゾルバは信頼のアンカーの類の鍵を得られる強固な仕組みをいくつか持つべき。		〇	
×	挙動をオーバーライドするオプションは存在しない	157	RFC4035		セキュリティ対応リゾルバはアトミックな単位で応答をキャッシュすべき。この単位は所有者名をもつRRSetとそのRRSetのDNSSEC RRを含む回答全体。		〇	
×	挙動をオーバーライドするオプションは存在しない	158	RFC4035		セキュリティ対応リゾルバはキャッシュのエントリーの中でどれか1つでも有効期限が来た時点でエントリー全体を破棄すべき。		〇	
×	挙動をオーバーライドするオプションは存在しない	159	RFC4035		セキュリティ対応リゾルバは問い合わせのADビットをクリアしなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	160	RFC4035		リゾルバは安全な方法で応答が得られない限りCD/ADビットを無視しなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	161	RFC4035		不良キャッシュを持つリゾルバは、検証に失敗したRRsetのTTLは自分で割り当てなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	162	RFC4035		不良キャッシュを持つリゾルバは、検証に失敗したRRSetのTTLをTTLの値は小さくすべき。（攻撃による認証失敗の場合、影響を小さくするため）		〇	

×除外 〇対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
×	挙動をオーバーライドするオプションは存在しない	163	RFC4035		一時的な認証失敗に影響されないよう、失敗の記録をもち一定閾値以上の失敗があった問い合わせに対してだけ不良キャッシュで応答すべき。		〇	
×	挙動をオーバーライドするオプションは存在しない	164	RFC4035		署名検証が必要となる条件のRRSet問い合わせに対して、不良キャッシュからRRSetを返してはならない。(たとえば逆にCD=1の問い合わせにたいしては不良キャッシュから返すべき)		〇	
×	挙動をオーバーライドするオプションは存在しない	165	RFC4035		有効な署名つきDNAMEからRFC2672の規定に従って署名なしCNAMEが合成された場合、DNAMEの署名をCNAMEも対象としているものとして処理しなければならない。(少なくともこの署名なしCNAMEを拒否してはならな		〇	
×	挙動をオーバーライドするオプションは存在しない	166	RFC4035		リゾルバはDNSKEYを次の方法で認証しなければならない。ゾーン頂点に起点となるDNSKEYが存在し、そのDNSKEY RRにゾーン鍵フラグがたっていること。DNSKEYのRRSIGが存在し、起点となるDNSKEYによってそのRRSIGを検証できること。(起点となるDNSKEYは別の仕組みによって取得されるものとする)		〇	5. DNS応答の認証
×	挙動をオーバーライドするオプションは存在しない	167	RFC4035		リゾルバは応答にDNSSECデータがないからといって認証情報が存在しないと解釈してはならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	168	RFC4035		リゾルバはゾーンに関する公開鍵が設定されている、またはゾーンの親から署名つきで委任にDSが付随している場合に、そのゾーンが署名つきであるとみなすべき。		〇	
×	挙動をオーバーライドするオプションは存在しない	169	RFC4035		セキュリティの島に対して外部からの手段で認証されたゾーン鍵を得られなかった場合、署名なしとみなした運用をすべき。		〇	
×	挙動をオーバーライドするオプションは存在しない	170	RFC4035		セキュリティ対応リゾルバは委任点における参照の応答においてDSもDSの不在を証明するNSECも含まれていない場合親ゾーンのネームサーバにたいしDSを問い合わせなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	171	RFC4035		セキュリティ対応リゾルバがDSの不在を証明する場合は親ゾーンのNSECを使用しなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	172	RFC4035		リゾルバが認証済みのDS RRSetに列挙されたアルゴリズムをそれもサポートしていない場合、子ゾーンを署名なしとみなして処理すべき。		〇	
×	挙動をオーバーライドするオプションは存在しない	173	RFC4035		セキュリティ対応リゾルバは以下の条件をすべて満たした場合にRRSIGと対象RRsetを認証可能 RRSIGとRRSetは同じ所有者名、同じクラスである RRSIGの署名者名フィールドがゾーンの名前である RRSIGの署名対象フィールドはRRSetのタイプと同じ。 RRSetの所有者名のラベル数がRRSIGのラベルフィールド以上 検証者の現在時刻がRRSIGの有効期間開始以上かつ有効期間終了以下 RRSIGの署名者名、アルゴリズム、鍵タグがゾーン頂点のDNSKEYにあるレコードと一致 上記条件に一致するDNSKEYがゾーン頂点のDNSKEY RRSetにあり、ゾーンフラグビットが設定されていなければならない		〇	
×	挙動をオーバーライドするオプションは存在しない	174	RFC4035		2つ以上のDNSKEYが認証可能条件に一致する場合、認証されるまで順番にDNSKEYを試し続けなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	175	RFC4035		RRSIGのラベル数フィールドがRRSetのFQDNのラベル数を超えていたら認証に使用してはならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	176	RFC4035	NSEC	セキュリティ対応リゾルバが親ゾーンから得られた委任に関するオリジナルのNSECを再構成する際に親側のNSEC RRを子側のNSEC RRと混同してはならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	177	RFC4035	NSEC	セキュリティ対応リゾルバが子ゾーンの頂点にあるオリジナルNSEC Rrsetを再構成する際に、子側のNSEC RRを親ゾーンのNSEC RRと混同してはならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	178	RFC4035		RRSIGのラベルフィールドがRRSetのFQDNのラベル数と一致しない場合、ワイルドカード展開が適切に行われたかどうかを検証しなければならない。		〇	

×除外 〇対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
×	挙動をオーバーライドするオプションは存在しない	179	RFC4035		リゾルバがRRSetを信頼できると判断した場合、検証者はRRSIGおよび認証済みRRSetの各TTLを、次の値の最小値を超えないように設定しなければならない。応答を受信した際のRRSetのTTL・応答を受信した際のRRSIG RRのTTL・RRSIG RRのオリジナルTTL・(RRSIGの有効期間終了時刻 − 現在時刻)		〇	
×	挙動をオーバーライドするオプションは存在しない	180	RFC4035		セキュリティ対応リゾルバがRRSetの不在を証明する場合には、ワイルドカードが一致するRRSetの不在も証明しなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	181	RFC4035		セキュリティ対応リゾルバは不在を証明するためのNSECが不足している場合、問い合わせを再送しなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	182	RFC4035		ただし、特定の問い合わせの回答だけの労力を注ぐことは抑制しなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	183	RFC4035	NSEC	認証済みNSECのタイプビットマップに関して、検証者はNSECおよびRRSIGビットは無視しなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	184	RFC4035		署名が検証できない場合、再帰検索サービスを提供するために検証しているのであれば、ネームサーバは問い合わせの元となったクライアントへの応答にRCODE=2(servfail)を返さなければならない。		〇	
×	挙動をオーバーライドするオプションは存在しない	185	RFC4035		ただし、オリジナルの問い合わせにCDビットが設定されている場合は完全な応答を返さねばならない。		〇	
×	スタブリゾルバの挙動はスコープ外	186	RFC4035		セキュリティ対応スタブリゾルバはDNSSEC RRタイプをサポートしなければならない。(すくなくともDNSSEC RRによって誤った処理をしてはならない)			スタブリゾルバに関する項目なので、適用除外
×	スタブリゾルバの挙動はスコープ外	187	RFC4035		検証機能つきセキュリティ対応スタブリゾルバはDOビットを設定しなければならない。			スタブリゾルバに関する項目なので、適用除外
×	スタブリゾルバの挙動はスコープ外	188	RFC4035		検証機能なしスタブリゾルバはアプリケーション層からの要求がない限りCDビットを立てた要求をすべきではない。			スタブリゾルバに関する項目なので、適用除外
×	スタブリゾルバの挙動はスコープ外	189	RFC4035		検証機能つきスタブリゾルバはCDビットを設定すべき。			スタブリゾルバに関する項目なので、適用除外
×	スタブリゾルバの挙動はスコープ外	190	RFC4035		セキュリティ対応スタブリゾルバは信頼できるセキュリティ対応ネームサーバから安全なチャネルでデータを得た場合以外はAD=1であっても代わりに行われた署名検証を信頼してはならない。			スタブリゾルバに関する項目なので、適用除外
×	スタブリゾルバの挙動はスコープ外	191	RFC4035		検証機能つきスバルリゾルバは応答にADビットが設定されていることを検査すべきではない。			スタブリゾルバに関する項目なので、適用除外
×	挙動をオーバーライドするオプションは存在しない	192	RFC5011		REVOKE bitが立ったDNSKEY RRSetを送出できること。		〇	
×	挙動をオーバーライドするオプションは存在しない	193	RFC5011		REVOKE bitが立ったDNSKEY RRSetが理解できること。		〇	
〇	named.confのmanaged-keysの設定で挙動が変化する	194	RFC5011		REVOKE bitが立っており,自分自身で署名されたDNSKEYを受信したリゾルバは、対応するDNSKEYをRevokeすること。		〇	
×	挙動をオーバーライドするオプションは存在しない	195	RFC5011		REVOKE bitが立っており,自分自身で以外の鍵で署名されたDNSKEYを受信したリゾルバは、対応するDNSKEYをRevokeしないこと。		〇	
〇	named.confのmanaged-keysの設定で挙動が変化する	196	RFC5011		RevokeされたDNSKEYはtrust anchorとしては使用できないこと。		〇	
×	挙動をオーバーライドするオプションは存在しない	197	RFC5011		RevokeされたDNSKEYはvalidate用途に使用できないこと(Revokeのvalidateを除く)。		〇	
×	挙動をオーバーライドするオプションは存在しない	198	RFC5011		リゾルバが正しく署名された新しいSEPの鍵を見た場合、Hold-down時間が経過しない限り新しい鍵は有効にならないこと。		〇	
×	挙動をオーバーライドするオプションは存在しない	199	RFC5011		リゾルバが正しく署名された新しいSEPの鍵を見た後、その新しい鍵のないDNSKEY RRSetに対する有効な署名を確認した場合、新しい鍵の受領プロセスを中止してタイマーがリセットされること。		〇	
×	挙動をオーバーライドするオプションは存在しない	200	RFC5011		新しいSEPの鍵を正しく署名するのに使われた鍵が、タイマーの期限の前に全てrevokeされた場合、新しい鍵の受領プロセスを中止してタイマーがリセットされること。		〇	

×除外 〇対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
〇	named.confのmanaged-keysの設定で挙動が変化する	201	RFC5011		タイマー期限が過ぎたら、次に新しい鍵によって正しく署名された DNSKEY RRSetを見たとき、新しい鍵はtrust anchorに追加されること。		〇	
〇	named.confのmanaged-keysの設定で挙動が変化する	202	RFC5011		タイマー期限来る前に、新しい鍵はtrust anchorに追加されていないこと。		〇	
×	挙動をオーバーライドするオプションは存在しない	203	RFC5011		タイマー期限以降に新しい鍵を含んだDNSKEY RRSetで正しく署名されているものを受領するまでは、新しい鍵はtrust anchorに追加されていないこと。		〇	
×	挙動をオーバーライドするオプションは存在しない	204	RFC5011		式 MAX(1時間, MIN(15日, DNSKEY のオリジナルTTL/2, RRSIGの有効期限間隔/2)) で表わされる間隔で設定されたtrust pointに対するクエリを行うこと。		〇	
×	挙動をオーバーライドするオプションは存在しない	205	RFC5011		trust pointに対するクエリに失敗した場合、式 MAX(1時間, MIN(1日, DNSKEY のオリジナルTTL*0.1, RRSIGの有効期限間隔*0.1)) で表わされる間隔でクエリを再送すること。		〇	
×	挙動をオーバーライドするオプションは存在しない	206	RFC5011		鍵の追加 Hold-down時間は30日か、trust pointの新しい鍵を含むDNSKEY RRSetのオリジナルTTLのどちらか大きい方であること。		〇	
×	挙動をオーバーライドするオプションは存在しない	207	RFC5011		鍵の削除 Hold-down時間(revoke後、clientが保持する時間)は30日であること。		〇	
×	挙動をオーバーライドするオプションは存在しない	208	RFC5011		リゾルバは一つのtrust pointについて少くとも5つのSEP鍵の管理ができること。		〇	
×	挙動をオーバーライドするオプションは存在しない	209	RFC5011		全てのtrust anchorがrevokeされたtrust pointは、trust pointでないものとして扱われること。		〇	
×	挙動をオーバーライドするオプションは存在しない	210	RFC5011		全てのtrust anchorがrevokeされたtrust pointは、その上位にtrust pointが存在しない場合、セキュアでないものとして扱われること。		〇	
×	挙動をオーバーライドするオプションは存在しない	211	RFC5115	NSEC3	NSEC3 RRが正しく送出できること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	212	RFC5115	NSEC3	NSEC3 RRが正しく認識できること。		〇	
×	挙動をオーバーライドするオプションは存在しない	213	RFC5115	NSEC3	NSEC3 RRのクラスはオリジナルの名前のクラスに一致すること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	214	RFC5115	NSEC3	NSEC3 RRのオリジナル TTLはSOA minimum TTLと一致すること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	215	RFC5115	NSEC3	NSEC3 RRのハッシュアルゴリズムが妥当なこと。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	216	RFC5115	NSEC3	NSEC3 RRのフラグが、0またはOpt-Outフラグのみであること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	217	RFC5115	NSEC3	NSEC3 のOpt-Outフラグがclearされている場合、unsigned delegationをcoverしないこと。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	218	RFC5115	NSEC3	NSEC3 のOpt-Outフラグがsetされている場合、unsigned delegationを除いた権威あるRRをcoverしないこと。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	219	RFC5115	NSEC3	NSEC3 のSalt Lengthは0～255であること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	220	RFC5115	NSEC3	NSEC3 のHash Lengthは1～255であること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	221	RFC5115	NSEC3	NSEC3 のNext Hashed Owner Name に対応する NSEC3 RRが存在すること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	222	RFC5115	NSEC3	Type bit map の window block 0のbit 0は0であること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	223	RFC5115	NSEC3	Type bit map の window block 0のbit 0をvalidatorは無視すること。		〇	

×除外 〇対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
×	挙動をオーバーライドするオプションは存在しない	224	RFC5115	NSEC3	Type bit map にtypeの存在しないwindow blockは含まれないこと。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	225	RFC5115	NSEC3	Type bit map の後側(trailing)のzero octetは取り除かれていること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	226	RFC5115	NSEC3	Type bit map の後側(trailing)はzero octetとして扱われること。		〇	
×	挙動をオーバーライドするオプションは存在しない	227	RFC5115	NSEC3	NSEC3PARAM RRが正しく送出できること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	228	RFC5115	NSEC3	NSEC3PARAM RRが正しく認識できること。		〇	
〇	NSEC3による署名を行わない場合、NSEC3PARAMが存在しない	229	RFC5115	NSEC3	ゾーンの頂点にNSEC3PARAM RRが存在すること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	230	RFC5115	NSEC3	NSEC3PARAM RRのFlagsは0であること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	231	RFC5115	NSEC3	NSEC3PARAM RRのHash algorithm、iterations, salt と同じ値のNSEC3 RRが、ゾーンの全てのhashed owner name に存在すること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	232	RFC5115	NSEC3	NSEC3PARAM RRのクラスは、対応するNSEC3のクラスと一致すること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	233	RFC5115	NSEC3	NSEC3PARAM RRのFlagsが0以外のものは無視されること。		〇	
×	挙動をオーバーライドするオプションは存在しない	234	RFC5115	NSEC3	権威のあるRRSetのオーナー名に対応するNSEC3 RRが存在すること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	235	RFC5115	NSEC3	RRの存在しない空の非ターミナルに対しては、それが署名されない委譲でありOpt-Out NSEC3 RRでcoverされる場合を除いてNSEC3 RRが存在すること。ドメイン名 a.b.c.d が存在する時、b.c.d, c.d, d等は非ターミナルである。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	236	RFC5115	NSEC3	NSEC3 RRのTTL値はSOA RRの最少TTL値に等しいこと。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	237	RFC5115	NSEC3	NSEC3 RRの type bit map は、全てのオリジナルのオーナー名にある全ての型に対応していること(NSEC3自身を除く)。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	238	RFC5115	NSEC3	応答が複数のNSEC3を含んでいる時は、全てのNSEC3 RRは同じハッシュアルゴリズム、同じiteration、および同じsalt値であること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	239	RFC5115	NSEC3	Name Error応答の場合は、QNAMEの非存在を証明するため、QNAMEのclosest encloser証明(最大2つのNSEC3 RR)およびclosest encloserにおけるwildcardをcoverする NSEC3 RRを返すこと。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	240	RFC5115	NSEC3	QTYPEがDS以外のNo Data応答の場合は、QNAMEに一致するNSEC3 RRを返すこと。このNSEC3 RRはQTYPEおよびCNAMEをType bit mapに含んでいないこと。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	241	RFC5115	NSEC3	QTYPEがDSの場合のNo Data応答の場合は、QNAMEに一致するNSEC3 RRがあればそれを返すこと。この時NSEC3 RRはDSおよびCNAMEをType bit mapに含んでいないこと。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	242	RFC5115	NSEC3	QTYPEがDSの場合のNo Data応答の場合は、QNAMEに一致するNSEC3 RRがなければQNAMEに対するclosest provable encloserの証明を返すこと。その証明のうちnext closerをcoverするNSEC3 RRはOpt-Out bitがsetされていること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	243	RFC5115	NSEC3	QTYPEがDSの場合のNo Data応答の場合は、serverがQNAMEにあるzone cutの両側に権威があるのであれば、zone cutの親側の証明が返されること。	〇	〇	

×除外 〇対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
×	挙動をオーバーライドするオプションは存在しない	244	RFC5115	NSEC3	QNAMEにwildcardが一致するが、QTYPEが存在しないNo Data応答の場合、QNAMEのclosest encloserの証明および、wildcard自身に一致するNSEC3 RRが返されること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	245	RFC5115	NSEC3	QNAME、QTYPEにwildcardが一致する場合、wildcard展開されたRRSetに加えてwildcardの直接の親の、(QNAMEについての)next closerをcoverするNSEC3 RRも返されること(wildcard以外にQNAMEにmatchするものがないことを証明する)。	〇	〇	
〇	非opt-out運用のゾーンでopt-outなしのNSEC3が帰ってくることを確認するべき	246	RFC5115	NSEC3	署名されない子ゾーンに対する問合せに対して委譲名に一致するNSEC3 RRが存在する場合、NSEC3 RRが応答に含まれること。またNSEC3 RRのbit mapにDSは存在していないこと。	〇		
×	重複して記載された項目なので除外	247	RFC5115	NSEC3	署名されない子ゾーンに対する問合せに対して委譲名に一致するNSEC3 RRが存在する場合、NSEC3 RRが応答に含まれること。またNSEC3 RRのbit mapにDSは存在していないこと。	〇	〇	246の重複
〇	opt-out運用のゾーンでopt-outのNSEC3が帰ってくることを確認するべき	248	RFC5115	NSEC3	署名されない子ゾーンに対する問合せに対して、Opt-Outのケースでは委譲名に一致するNSEC3 RRが存在しない場合がある。この場合はclosest provable encloserの証明が応答に含まれていること。この証明のnext closerをcoverするNSEC3 RRは Opt-Outがセットされていること。	〇		
×	挙動をオーバーライドするオプションは存在しない	249	RFC5115	NSEC3	問合せのQNAMEが存在するNSEC3 RRのオーナー名であって、QNAMEにもその子孫にもRRが存在しない場合、その問合せはName Error応答を返すこと。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	250	RFC5115	NSEC3	存在しないQNAMEのハッシュが、存在するNSEC3 RRのオーナー名と衝突した場合は、サーバはRCODE 2(server failure)を返すこと。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	251	RFC5115	NSEC3	サーバは未知のハッシュアルゴリズムを見付けた場合はゾーンのロード時に拒絶し、問い合わせに対してはRCODE 2(server failure)を返すこと。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	252	RFC5115	NSEC3	Dynamic Updateにより名前が削除される場合、その名前による空の非ターミナルのNSEC3 RRは削除されていること(他の名前によりその空の非ターミナルが必要とされる場合を除いて)。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	253	RFC5115	NSEC3	Dynamic Updateにより名前が追加される場合、その名前による空の非ターミナルに対応するNSEC3 RRが追加されていること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	254	RFC5115	NSEC3	Dynamic UpdateによりNSEC3 RRが削除される時、(NSEC3 連鎖による)その前のNSEC3 RRの Opt-Out フラグは変更されないこと。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	255	RFC5115	NSEC3	Dynamic UpdateによりNSEC3 RRが追加される時、その Opt-Out フラグは以前その名前をcoverしていたNSEC3 RRのものであること。	〇	〇	
×	挙動をオーバーライドするオプションは存在しない	256	RFC5115	NSEC3	未知のハッシュ型のNSEC3 RR はにせものとして無視されること。		〇	
×	挙動をオーバーライドするオプションは存在しない	257	RFC5115	NSEC3	0か1以外のフラグを持つNSEC3 RRはにせものとして無視されること。		〇	
×	挙動をオーバーライドするオプションは存在しない	258	RFC5115	NSEC3	validatorがclosest encloserを発見したら、オリジナルのオーナー名をclosest encloserとして持つNSEC3 RRのゾーンが適切か(親ゾーンのものか)をチェックすること。		〇	
×	挙動をオーバーライドするオプションは存在しない	259	RFC5115	NSEC3	Name Error応答があった場合、validatorはQNAMEに対するclosest encloser証明と、closest encloserにおけるwildcardをcoverするNSEC3 RRを検証すること。		〇	
×	挙動をオーバーライドするオプションは存在しない	260	RFC5115	NSEC3	No Data Responce応答があり、QTYPEがDSでない場合、validatorはQNAME に一致するNSEC3 RRが存在し、QTYPEおよびCNAMEがType Bit Mapににないことを検証すること。		〇	
×	挙動をオーバーライドするオプションは存在しない	261	RFC5115	NSEC3	No Data Responce応答があり、QTYPEがDSの場合かつQNAME に一致するNSEC3 RRが存在する場合は、validatorはDSおよびCNAMEがType Bit Mapににないことを検証すること。		〇	

×除外 〇対象	除外した理由/しなかった理由	No.	該当RFC	条件	確認項目	権威サーバ	フルリゾルバ	備考
×	挙動をオーバーライドするオプションは存在しない	262	RFC5115	NSEC3	No Data Responce応答があり、QTYPEがDSの場合かつQNAME に一致するNSEC3 RRが存在しない場合は、validatorはQNAMEに対するclosest provable encloserの証明が応答に存在し、next closerをcoverするNSEC3 RRのOpt-Out bitがsetされていることを検証すること。		〇	
×	挙動をオーバーライドするオプションは存在しない	263	RFC5115	NSEC3	QNAMEに対するclosest encloser証明を検証し、closest encloserにアスタリスク ラベルを追加した名前によって生成されるwildcard名に一致するNSEC3 RRの存在を検証しなければならない。さらにQTYPEとCNAMEのbitが対応するNSEC3 RRに存在しないことを検証すること。		〇	
×	挙動をオーバーライドするオプションは存在しない	264	RFC5115	NSEC3	validatorはQNAMEに対するwildcardが応答された場合、応答に存在するQNAMEに対応する next closer をcoverするNSEC3 RRを検証すること(QNAMEそのものは存在せず、正しいwildcardは存在することを検証す		〇	
×	挙動をオーバーライドするオプションは存在しない	265	RFC5115	NSEC3	委譲名に一致するNSEC3 RRが応答に存在する場合、type bit map にNS bitがセットされておりDS bitがセットされていないことを検証すること。		〇	
×	挙動をオーバーライドするオプションは存在しない	266	RFC5115	NSEC3	委譲名に一致するNSEC3 RRが応答に存在する場合、NSEC3 RRが親ゾーンのものであることを検証すること(すなわち、SOA bit が type bit mapにセットされていないこと)。		〇	
×	挙動をオーバーライドするオプションは存在しない	267	RFC5115	NSEC3	委譲名に一致するNSEC3 RRが存在しない場合、委譲名のclosest provable encolser証明を検証すること。また委譲名のnext closerをcoverするNSEC3 RRはOpt-Out ビットがセットされていることを検証すること。		〇	
×	挙動をオーバーライドするオプションは存在しない	268	RFC5115	NSEC3	cache resolverは応答時に適切なNSEC3 RRを検索できること。		〇	
×	挙動をオーバーライドするオプションは存在しない	269	RFC5115	NSEC3	next closer をcoverするNSEC3 RRがOpt-Outビットがセットされている場合、closest (provable) encloser証明が含まれている応答を返すならばADビットがsetされていないこと。		〇	
×	dnssec-signzoneはIterationをオプション指定できるが、このIterationを内部で制限しているので運用上の問題になりえな	270	RFC5115	NSEC3	各々の鍵長に対して、以下のサイズ越えるIterationを使用していないこと(1024鍵長/150回,2048鍵長/500回, 4096鍵長/2500回)。	〇	〇	