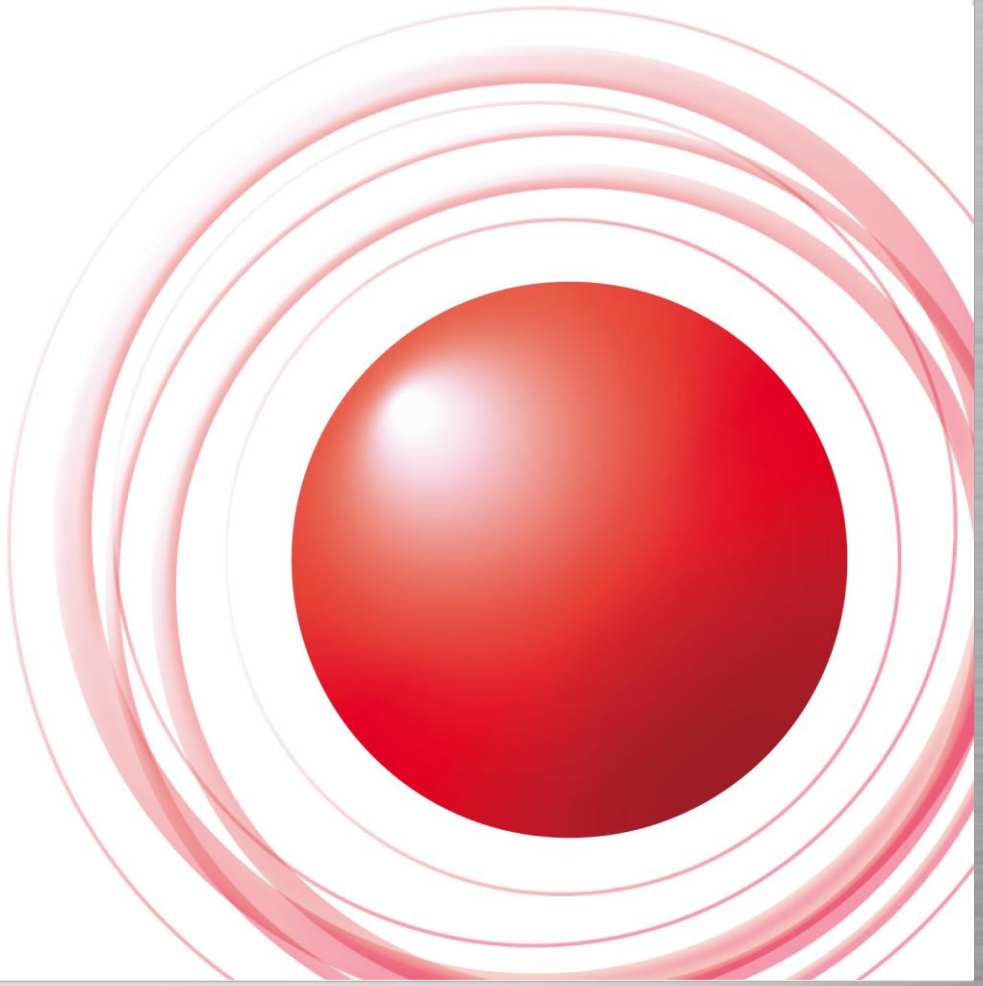
RFC5155
DNS Security (DNSSEC) Hashed Authenticated Denial of Existence

IIJ Internet Initiative Japan



2010/8/23

(株) インターネットイニシアティブ
島村 充 <simamura@iij.ad.jp>
高橋 辰徳 <tatsu@iij.ad.jp>
鈴木 高彦 <takahiko@iij.ad.jp>

Ongoing Innovation

Abstract

- DNSSEC拡張で不在証明のためにNSEC RRが導入された
- 本RFCではNSEC3という、同様に不在を証明するためのRRを導入する
- またNSEC3ではzoneのRRを収集されてしまうことも防ぐ

1. Introduction
1.1. Rationale

- **DNSSECではNSECという不在証明のRRを導入した**
  - 副作用としてゾーンの中身の列挙を許してしまう
  - これは望まれないポリシーの課題となる
- **NSECレコードの例：**

```
a.example.jp. IN NSEC c.example.jp. A RSIG NSEC
```

  - → b.example.jpは無い(ということを表す)
- **これを再帰的に繰り返せばゾーンのRRのリストが作成可能**
- **NSECにより生じる問題点**
  - spam送信のためのアドレスリストのソースになる
  - レジストリによっては登録者の情報を隠していたり、ゾーンのコピーを禁止している
- **NSECを導入するとこのような問題が顕在化する**
- **別の問題**
  - 未署名のゾーンに対して暗号的に安全な委任をするコストが高い

    → Opt-Outを使うのが適切である
- **これらを解決するためにNSEC3 RRを導入する**

1.3. Terminology

- **参照するべきRFC**
  - RFC1034, RFC1035, RFC4033, RFC4034,
  - RFC4035, RFC2136, RFC2181, RFC2308
- **用語**
  - Zone enumeration
    - 連続したクエリによってゾーンの全内容を取得すること
  - Original owner name
    - hashed owner nameに対応した、owner name
  - Hashed owner name
    - owner nameにハッシュ関数を適用して作られたowner name
    - `hash(origilanl owner name) = hashed owner name`
  - Hash order
    - hashed owner nameをその数値順に並べたもの
    - base32でエンコードされているならばRFC4034で規定された Canonical DNS Name Order(Section 6.1.)と同じ順序になる
  - Empty non-terminal
    - RRを持っていないが1つ以上のサブドメインがあるドメイン名

1.3. Terminology (cont.)

- Delegation
  - 現在のzone apexとは異なる名前を持ったNS RRセット。子ゾーンへの委任を表す
- Secure Delegation
  - Delegationと署名されたDS RRセットを含む名前。署名された子ゾーンへの委任を表す
- Insecure delegation
  - Delegationを含んでいるが、DS RRセットが不足している名前。署名されていない子ゾーンへの委任を表す。
- Opt-Out NSEC3 resource record
  - Opt-Outフラグが1にセットされたNSEC3 RR
- Opt-Out zone
  - 少なくとも1つのOpt-Out NSEC3 RRを持ったゾーン
- Closest encloser
  - 最も長い、存在する名前の先祖。RFC4592 Secion 3.3.1.参照
  - 合成されていないドメイン名のうち最長の物
  - ワイルドカードレコードは “*.<closest encloser>”となる

1.3. Terminology (cont.)

- Closest provable encloser
  - 存在を証明することが可能な、最も長い名前の先祖
  - Closest encloserとはOpt-Out zoneにあるかどうかの違いだけ
- Next closer name
  - closest provable encloserより長い名前
- Base32
  - RFC4648で定義されている"Base 32 Encoding with Extended Hex Alphabet"の事
  - 末尾にパディングで使う”=”はNSEC3では使わない
  - 0～9→そのまま, 10～31→A to V
- To cover
  - 名前のハッシュか”next closer”な名前がowner nameとnext hashed owner nameの間にあるならNSEC3 RRは名前を”cover”していると言われる
- To match
  - NSEC3 RRのowner nameがハッシュ化されたowner nameと同じならNSEC3 RRは名前に”match”すると言われる

2. Backwards Compatibility

- **上書きされるRFC**
  - RFC4033, RFC4034, RFC4035
- **RFC5155に未対応なsecurity-awareリゾルバ(NSEC3-unawareリゾルバ)は検証に失敗するようになる**
- **NSEC3-unawareリゾルバがNSEC3で署名されたゾーンからのDNS応答を検証しないようにするためにシグナルテクニックを用いる**
  - 新しいDNSKEYアルゴリズム識別子を割り当てる
    - アルゴリズム6(DSA-NSEC3-SHA1): アルゴリズム3(DSA)のエイリアス
    - アルゴリズム7(RSASHA1-NSEC3-SHA1): アルゴリズム5(RSASHA1)のエイリアス
- **ゾーンがDNSKEY　RRに署名するときは前述の2つのアルゴリズムのみを用いて署名しなくてはならない(MUST)**
  - 2つの識別子がNSEC3-unawareリゾルバにとって未知の物で、NSEC3で署名されたゾーンからのDNS応答を安全でないと扱うから (RFC4035 Secion 5.2参照)

2. Backwards Compatibility (cont.)

- 2つのアルゴリズムはSHA1を用いている。別のハッシュアルゴリズムを使いたいときは新しいエイリアスを割り当てる(Section 12.1.3)
- NSEC3に対応しているSecurity awareリゾルバは新しいアルゴリズム識別子を認識しなければならない(MUST)
- そのエイリアスが指し示す先のアルゴリズムと同等に扱わなくてはならない(MUST)
- NSEC3への移行はSecion 10.4.で述べる

3. The NSEC3 Resource Record

- **NSEC3 RRはDNS RRセットの不在証明を提供**
- **NSEC3 RRはNSEC3 RRのoriginal owner nameに存在するRRタイプを列挙する**
  - ゾーンのhash orderに含まれるnext hashed owner nameを含む
  - ゾーン中のNSEC3 RRの完全なセットはRRのoriginal owner nameのためにRRセットが存在することを示し、ゾーン内の hashed owner nameのチェインを形成する
  - zone enumerationを防ぐためにoriginal owner nameをハッシュ化したowner nameでなくてはならない
- **NSEC3 RRが表しているもの**
  - ハッシュを構築するために
    - どのハッシュ関数をつかったか
    - どのsaltを用いたか
    - 何回繰り返したか

3. The NSEC3 Resource Record (cont.)

- **ハッシュ化のテクニックについてはSecion 5**
- **未署名なdelegationのhashed owner nameはチェインから除外されるべき**
  - owner nameや未署名なdelagationの”next closer”な名前のハッシュはOpt-Out NSEC3 RRとして参照され、フラグの存在として表される
- **NSEC3 RRのowner nameはゾーンの名前を単一のラベルとして付加したハッシュ化されたowner nameをbase32でエンコードしたものである**

```
owner name = base32(hash(<owner name>.example.jp))
```

- **NSEC3 RRのtype valueは”50”**
- **NSEC3 RRのRDATAフォーマットはクラス非依存**
- **クラスはoriginal owner nameのクラスと同じでなくてはならない (MUST)**
- **NSEC3 RRはSOAのminimum TTLフィールドと同じTTL値を持つべき (SHOULD)**
  - これはnegative cache (RFC2308)の考えから

3.1. RDATA Fields

- **3.1.1. Hash Algorithm**
  - ハッシュ値を構築するのに使われる暗号的なハッシュアルゴリズムを特定するフィールド
  - Section 11.で規定しているNSEC3 hash algorithm registryで定義されている
- **3.1.2. Flags**
  - 8個の1bitのフラグから成る
  - 異なる処理を示すのに使うことができる
  - すべての未定義のフラグはゼロでなくてはならない
  - 唯一定義されたフラグはOpt-Outフラグである
- **3.1.2.1. Opt-Out Flag**
  - Opt-Outフラグがセットされていたら、NSEC3レコードは0個以上の未署名のdelegationをカバーする
  - セットされていないならNSEC3レコードは1個も未署名のdelegationをカバーしない
    - NSEC3 RRが未署名のdelegationをカバーすべきかどうかを示す
  - Opt-OutフラグはFlagsフィールドの最下位ビットである
  - Opt-Outフラグの使い方はSection 6.で述べている

3.1. RDATA Fields (cont.)

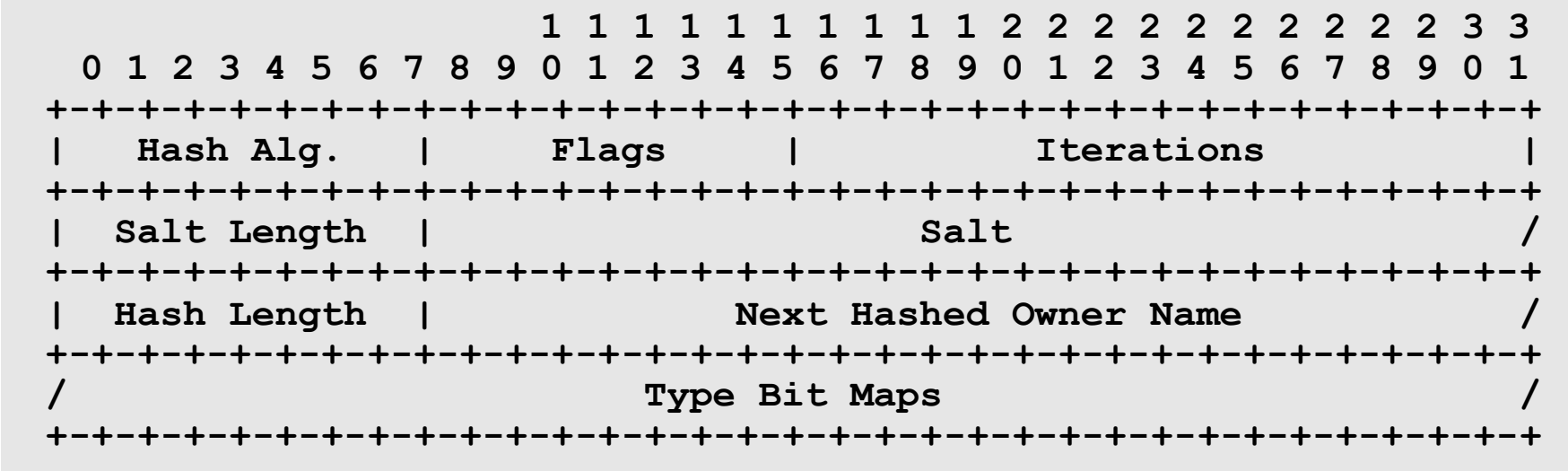
- **3.1.3. Iterations**
  - ハッシュ関数が実行される回数を定義
  - 複数回実行すると辞書攻撃に対する耐性を得る
  - しかし、サーバ・リゾルバともに計算量が増える
  - このフィールドの使い方の詳細はSecion 5., 上限値はSection 10.3.
- **3.1.4. Salt Length**
  - Saltフィールドの長さをオクテットで表し、値は0～255の間
- **3.1.5. Salt**
  - 事前計算された辞書攻撃に対抗するために、ハッシュ値を求める前にoriginal owner nameに付加される
    - どのように使われるかはSection 5.
- **3.1.6. Hash Length**
  - Next Hashed Owner Nameフィールドの長さを定義していて、値は1～255オクテットの間

3.1. RDATA Fields (cont.)

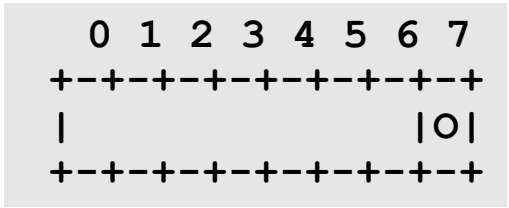
- **3.1.7. Next Hashed Owner Name**
  - ハッシュの順序でnext hashed owner nameを含む
  - この値はバイナリフォーマット
  - Next Hashed Owner Nameフィールドは規定のNSEC3 RRのowner nameのハッシュを含んでいる
  - ハッシュの順序でゾーン中で最後のNSEC3 RRのNext Hashed Owner Nameの値は一番始めのNSEC3 RRのhashed owner nameと同じになる
  - NSEC3 RRのowner nameとは違って、このフィールドの値は追加ゾーン名を含まない。
- **3.1.8. Type Bit Maps**
  - NSEC3 RRのoriginal owner nameに存在するRRセットのtypeを特定する

3.2. NSEC3 RDATA Wire Format

- NSEC3 RRのRDATAは以下のように表される



- Hash Algorithm: 1オクテット
- Flagsフィールド:　1オクテットで、Opt-Outフラグは以下の通り最下位ビット



3.2. NSEC3 RDATA Wire Format (cont.)

- **Iterations: 16bitの符号無し整数で、最上位ビットが先頭**
- **Salt Length: 符号無しオクテット。もし値が0ならば次のSaltフィールドは省略される**
- **Salt: 存在するならばバイナリオクテットのシーケンスとしてエンコードされる。長さはSalt Lengthで定義**
- **Hash Length: 符号無しオクテット。Next Hashed Owner Nameの長さをオクテットで表したもの**
- **Next Hashed Owner Name: base32でエンコードされたものではなく、ハッシュのバイナリ値そのもの。**
  - それを含むゾーンの名前は含まれていない。長さはHash Lengthで定義
- **3.2.1. Type Bit Maps Encoding**
  - RFC4034 Secion 4.1.2.で規定
  - ここでは省略

3.3. Presentation Format

- **RDATA部の各フィールドの表現記法**
  - Hash Algorithm: 符号無し10進数。最大値255
  - Flags: 符号無し10進数。最大値255
  - Iterations: 符号無し10進数。0～65535の範囲内
  - Salt Length: 表現されない
  - Salt: case insensitiveな16進数のシーケンス
    - ホワイトスペース不可
    - Salt Lengthが0の時は”-”として表す
  - Hash Length: 表現されない
  - Next Hashed Owner name: case insensitiveでホワイトスペースのないbase32のunpaddedなシーケンスとして表される
  - Type Bit Maps: RRタイプのニーモックのシーケンスとして表される。ニーモックが未知の場合RFC3597のSection 5.で定義されたタイプの表現を用いなくてはならない (MUST)

4. The NSEC3PARAM Resource Record

- **(Hash Algorithm, Flags, Iterations, Saltなど)hashed owner nameを計算する権威サーバによって必要とされるNSEC3のパラメータを含むRR**
- **zone apex中にNSEC3PARAM RRがあることは特定のパラメータが権威サーバが否定的なレスポンスの際にNSEC3 RRのセットを適切に選ぶために使われるべきであることを示す**
- **NSEC3PARAM RRはバリデータ・リゾルバによっては使われない**
- **NSEC3PARAM RRがゾーンのapexにFlagsフィールドの値が0で存在する**

  → 同じHash Algorithm, Iterations, Saltのパラメータを使ったNSEC3 RRがzone中のすべてのhashed owner nameに対して存在しなくてはならない (MUST)

  = zoneには同じHash Algorithm, Iterations, Saltのパラメータを持ったNSEC3 RRの完全なセットが含まれていなくてはならない(MUST)ことを表す

- **NSEC3PARAM RRのowner nameはzone apexの名前である**
- **NSEC3PARAM RRのtype valueは51**

4. The NSEC3PARAM Resource Record (cont.)

- **NSEC3PARAM RRのRDATAフォーマットはクラスに非依存で以下で述べるとおり**
- **NSEC3PARAM RRのクラスはこのRRが参照しているNSEC3 RRと同じクラスでなくてはならない(MUST)**
- **4.1., 4.3.は3.1., 3.3.と同じなため省略**
  - FlagsフィールドのOpt-Outフラグは使われずゼロ
  - Flagsフィールドがゼロでない場合はNSEC3PARAM RRは無視されなくてはならない (MUST)
- **4.2. NSEC3PARAM RDATA Wire Format**

```
  1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   Hash Alg.   |     Flags     |           Iterations          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Salt Length  |                        Salt                   /
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5. Calculation of the Hash

- **以下の3つのNSEC3 RDATAフィールドを用いてハッシュを計算**
  - Hash Algorithm
  - salt
  - iterations
- **ハッシュの計算方法**
  - NSEC3 RRで指定された Hash Algorithm を用いる
    - H(x): xのハッシュ
    - k: Iterations
  - ハッシュの計算は以下の通り
    - IH(salt, x, 0) = H(x || salt), and
    - IH(salt, x, k) = H(IH(salt, x, k-1) || salt), if k > 0

5. Calculation of the Hash (cont.)

- **owner name のハッシュの計算は以下の通り**
  - IH(salt, owner name, iterations)
- **owner name はcanonical formである**
- **canonical formは以下の通り**
  1. FQDNであること (名前の圧縮がされていないこと)
  2. 大文字はすべて小文字に変換する
  3. owner name がワイルドカード の場合、ワイルドカード(*)は展開しない

6. Opt-Out

- **Opt-Out の存在理由**
  - **NSEC の場合の動作**
    - NS のレコードの委譲の分だけ全ての NSEC レコードを作らなければならない
    - 例えばレジストリのように非常に多くのドメインを委任しているところは下位ドメインのDNSSECの対応に関わらず、全てのNSEC レコードとその RRSIG を作らなければならない
    - 非常に高コストになる
  - **NSEC3 の場合の動作**
    - 委任の NS レコードがあるだけでは NSEC3 レコードは作らない
    - 委任の NS レコードに加えて DS レコードがセットで登録されている場合にのみ NSEC3 レコード及び RRSIG を作る
    - DNSSECに対応してる子ゾーンだけでよいのでコストが低い
- **Opt-Out を使うときのデメリット**
  - 特になし

    (※ 委譲している NS で ＤＳ がないものの不在証明は暗号学的に不完全で証明されていない)

6. Opt-Out

- **Opt-Outの仕様**
  - delegation pointでのNS RRSetは署名されておらず、DS RRSetも同様に署名されない ([RFC4033]、[RFC4034]、[RFC4035])
  - Opt-Out flagが立っていない状態では、子ゾーンのセキュリティ状態がDS RRSet の有無によって決定される
  - DS RRSetの存在/非存在は署名されたNSEC3 RRSetによって暗号学的に証明される
  - Opt-Out flagを立てることによって署名付きzoneの中でNSEC3 RRを使わずに、安全でないdelagationを許可する
  - ハッシュ化したowner nameもしくはdelegation pointのnext closer nameがNSEC3 RRのowner nameと次にハッシュ化されたowner nameとの間にあるならばOpt-Out flagの立ったNSEC3 RRは委譲をcoverしている
  - Opt-Out flagの立ったNSEC3 RR はそれがcoverしている範囲に安全でない委譲の有無を示すことはできない
  - これにより、再署名や再計算無しにNSEC3 RR の連鎖にNSEC3 RRを挿入や削除を行うことができる

6. Opt-Out (cont.)

- Opt-Out flagの立ったNSEC3 RRは他のauthoritative RRSetの存在/非存在を示すことができる
- Opt-Out flagの立っている NSEC3 RR は安全でない委譲と元が同じであるようなowner name を持ってもよい(MAY)
  - この場合、タイプマップでDSビットが立っていないことによって、安全でない委譲であることが示される。また、署名されたNSEC3 RRによってdelegationの存在が示される
- Opt-Out を使用している zone は Opt-Out flagの立ったNSEC3 RRs と、Opt-Out flagの立っていないNSEC3 RRを同時に含む(MAY)
- Opt-Out flagの立っていないNSEC3 RRの場合、自身とnext hashed owner nameで示されるRRの間に、いかなる安全でないdelegationもあってはならない(MUST NOT)
- Opt-Out flagが立っているなら、hashed owner nameもしくは安全でないdelegationのhashed next closer nameのみをcoverするものではなくてはならない(MUST)

7. Authoritative Server Considerations

- **7.1. Zone Signing**
  - **NSEC3 を使用するゾーンは以下の特性を満たす必要がある**
    - authoritativeなRRSetsがあるゾーンの中のそれぞれのowner nameには、対応するNSEC3 RRがなければならない。(MUST)

      未署名なdelegationに対応するowner nameは対応する NSEC3 RR を持ってもよい。 (MAY)

      対応するNSEC3 RRがない場合、delegationのnext closer nameをcoverするOpt-Out flagの立ったNSEC3 RRが存在しなければならない。 (MUST)

      他のnon-authoritativeなRRs は対応する NSEC3 RRを持たない
    - empty non-terminal は対応するNSEC3 RRを持たなければならない(MUST)

      ただし、empty non-terminalがOpt-Out flagの立ったNSEC33 RRによってcoverされる、安全でないdelegation の場合はこの限りではない
    - NSEC3 RR の TTL 値はゾーンの SOA RR の最小のTTL 値と同じにすべき(SHOULD)
    - NSEC3 RR のみ存在する場合を除き、署名ゾーンの全てのNSEC3 RR の Type Bit Maps フィールドはオリジナルの owner name で存在するすべてのRRのタイプの存在を示さなければならない。(MUST)

      NSEC3 のタイプ自体が Type Bit Maps に存在しないことを意味していることに注意

7. Authoritative Server Considerations (cont.)

- **NSEC3 RRs の適切な構成手順 (※方式はこれだけではない)**
  1. ハッシュアルゴリズム, salt, iterationを選択
  2. ゾーンのoriginal owner name それぞれに NSEC3 RR を追加
     - Opt-Outを使うならば、署名されていないdelegationの owner name は除いてもよい (MAY)
     - NSEC3 RRはoriginal owner nameのhashであり、zone nameの前にラベルとして追加される
     - Next Hashed Owner Name の値はここでは空白である
     - Opt-Outを使うなら、Opt-Out bitを立てる
     - 衝突検出目的のために必要に応じてNSEC3 RRのoriginal owner nameを記録する
     - さらに、衝突検出目的のために、必要に応じてワイルドカードを先頭につけたoriginal owner nameのNSEC3 RRを作成する。このNSEC3 RRは一時的なものとする

7. Authoritative Server Considerations (cont.)

3. オリジナル owner nameそれぞれのRRに対して、Type Bit Mapsフィールドの対応するビットをセットする
4. zone apexとoriginal owner name の間のラベルの数の違いが1以上あるなら、zone apexとoriginal owner name の間の各empty non-terminalに対してNSEC3 RRの追加が必要である

   これにより重複したhashed onwer nameを持つNSEC3 RRsを生成するかもしれない

   衝突検出のために、Step 1と同様にワイルドカードの衝突を検出するために作成した一時的なNSEC3 RRとoriginal owner nameのNSEC3 RRを必要に応じて記録しておく
5. NSEC3 RRs のセットをハッシュの順番でソートを行う
6. hashed owner nameが同一のNSEC3 RRを、それらのRR typeの集合をtype bit map fieldに持つような単一のNSEC3 RRに置き換える

   もし、original owner name を記録しているならば、すべてのマッチするNSEC3 RRは同一のoriginal owner nameに由来するので、まとめたときに衝突を検出できる。一時的なNSEC3 RRはすべて捨てる

7. Authoritative Server Considerations (cont.)

7. 各 NSEC3 RR では、ハッシュの順番で次にくるNSEC3 RRを使用し、次のhashed owner name を挿入する

   zone で最後の NSEC3 RR の次のhashed owner name はハッシュの順番で並べたときに最初の NSEC3 RR のhashed owner name となる
8. 最後に上記のプロセスで用いたのと同じHash Algorithm, Iterations, Salt を持つNSEC3 PARAM RRをzone apexに加える

※ ハッシュの衝突が検出された場合、新しい Salt を選択して署名のプロセスをやりなおす

7. Authoritative Server Considerations (cont.)

- **7.2. Zone Serving**
  - この仕様は権威サーバが生成する、DNSSECに従ったDNS応答を変更する

    特にその応答における NSEC RRs の使用を NSEC3 RRsに置き換える
  - 以下の応答の場合では、DNSSEC [RFC4035] で要求されたNSEC RRs は NSEC3 RRs に置き換えられる

    NSEC RRs を含まない応答はこの仕様による仕様の変更はない
  - 複数の NSEC3 RRs を含む応答を返すとき、NSEC3 RRs のすべてが同じハッシュアルゴリズム, iterations, およびSaltの値を使用しなければならない(MUST)

    Flagsフィールドの値は 0 か 1 のどちらかでなければならない(MUST)

7. Authoritative Server Considerations (cont.)

- **7.2.1. Closest Encloser Proof**
  - 以下はQNAME のancestorが QNAME の closest encloser であるという証明である
  - この証明は 2 つ以上の異なる NSEC3 RRs から成る:
    - closest (provable) encloserにマッチする NSEC3 RR
      - closest encloserの候補を提示し、具体的な encloser が実際に存在することを証明
    - closest encloser の"next closer" nameをcoverするNSEC3 RR
      - closest encloser の候補が最も近いことを証明
      - QNAME(とQNAMEとclosest encloserの間にいかなるancestor)が存在しないことを証明
  - 上記の NSEC3 RRs は「closest encloser proof」として説明の中で参照される

7. Authoritative Server Considerations (cont.)

- 例えば “alpha.beta.gamma.example” というowner name の非存在を示す closest encloser proof は “gamma.example” が closest encloserであることによって証明されるだろう。
- この応答には “gamma.example” にマッチする NSEC3 RR と “beta.gamma.example” をカバーする NSEC3 RR が含まれる
- ※ “beta.gamma.example.” は “next closer” nameである
- Opt-Outの場合、Opt-Outでカバーされたinsecureなdelegation自体、もしくは一部なので、実際にclosest encloserであることを証明できないかもしれない
- この場合、実際の closest encloser を証明することの代わりに、closest provable encloser が用いられる
- すなわち closest enclosing authoritative name が代わりに使用される
- この証明に使用されるNSEC3 RRのセットは「closest provable encloser proof」と呼ばれる

7. Authoritative Server Considerations (cont.)

- **7.2.2. Name Error Responses**
  - QNAME が存在しないことを証明するためには、closest encloser proof と closest encloser において(存在しない)wildcard RR をカバーしている NSEC3 RR がレスポンスに含まれていなければならない(MUST)

    (最大)3つのNSEC3 RRの集合はQNAMEが存在しないことと QNAMEにマッチするワイルドカードが存在しないことの両方を証明する
  - (例)

    “gamma.example“ が QNAME の closest provable encloser であるなら、”*.gamma.example.” をカバーしているNSEC3 RRが応答の authority section に含まれる
- **7.2.3. No Data Responses, QTYPE is not DS**
  - サーバはQNAMEにマッチする NSEC3 RR を含まなければならない (MUST)

    NSEC3 RRはQTYPEもしくはCNAMEのどちらかに対応するビットを自身のタイプビットマップフィールドにもってはいけない (MUST NOT)

7. Authoritative Server Considerations (cont.)

- **7.2.4. No Data Responses, QTYPE is DS**
  - QNAME にマッチする NSEC3 RR がある場合：
    - サーバはそのNSEC3 RRを返さなければならない(MUST)

  DSとCNAMEに対応するビットを、このNSEC3 RR のタイプビットマップフィールド にセットしてはならない(MUST NOT)

  QNAME にマッチする NSEC3 RR がない場合：

  サーバはそのQNAMEのclosest provable encloser proof を返さなければならない (MUST)

  “next closer” name をカバーする NSEC3 RR は Opt-Out bit を立てなければならない(MUST)

  (これは定義によるものなので間違いない -- もし、Opt-Out bit がセットされていと、不都合 が生じるだろう)

  もし、サーバがQNAMEのzone cutの両側で権威があるなら、zone cutの親側のproofを返さなければならない(MUST)

7. Authoritative Server Considerations (cont.)

- **7.2.5. Wildcard No Data Responses**
  - QNAME にマッチするワイルドカードがあるが、その名前に対してQTYPEが存在しないならば、レスポンスは QNAME のclosest encloser proof を含まなければならない(MUST), そしてワイルドカードにマッチする NSEC3 RR を含まなければならない(MUST)

    この組み合わせは QNAME 自体が存在しないことと、QNAMEにマッチするワイルドカードが存在しないことの両方を証明する

    QNAME の closest encloser は ワイルドカード RR の直接のancestorでなくてはならない(MUST)ことに注意
- **7.2.6. Wildcard Answer Responses**
  - QNAMEとQTYPEにマッチするワイルドカードがあるなら、応答のanswer sectionでワイルドカードにマッチしたRRSetとそれにマッチしたという証明を返さなくてはならない

    この証拠はQNAMEが存在しないことと QNAME の closest encloser とワイルドカードの直接のancestorが同じでないことの両方をを証明することによって示される

    (例) 正しいワイルドカードにマッチした)

7. Authoritative Server Considerations (cont.)

- 最後に、ワイルドカードの直接のancestorのnext closer nameをカバーするNSEC3 RRを返さなくてはならない(MUST)
- closest encloserにマッチするNSEC3 RRは返す必要はない、なぜならこのclosest encloserの存在は応答中の展開されたワイルドカードの存在によって示されるからである

- **7.2.7. Referrals to Unsigned Subzones**
  - delegation name にマッチする NSEC3 RR があれば、NSEC3 RR がレスポンスに含まれなければならない(MUST)

    NSEC3 RR の type bit maps の DS bit はセットしてはならない(MUST NOT)
  - zone が Oput-Out であるなら、delegation に対応する NSEC3 RR は無い かもしれない

    closest provable encloser proof は応答に含まれていなければならない(MUST)

    delegation の“next closer” name をカバーする前述のNSEC3 RRは Opt-Out フラグが立っていなければならない(MUST)

7. Authoritative Server Considerations (cont.)

- **7.2.8. Responding to Queries for NSEC3 Owner Names**
  - NSEC3 RRs の owner name は他の owner name のように NSEC3 RR の連鎖に含まれない

    結果として、各NSEC3 owner nameは別の NSEC3 RR でカバーされており、NSEC3 RR の存在を効果的に否定している

    NSEC3 RRの存在は、そのRRSIG RRSetによって示されるので、矛盾しているようにみえる
  - 以下の条件がすべて当てはまる場合 :
    - QNAME は存在する NSEC3 RR の owner name と等しい
    - QNAMEにもその子孫にも、いかなるRR typeも存在しない
  - Name Error 応答 (セクション7.2.2) を返さなければならない(MUST)

    もしくは NSEC3 RR の owner name が存在していないかのように権威サーバは振る舞う
  - NSEC3 RRs が AXFR や IXFR クエリの応答となることに注意

7. Authoritative Server Considerations (cont.)

- **7.2.9. Server Response to a Run-Time Collision**
  - 存在しない QNAME のハッシュが 存在する NSEC3 RR の owner name と衝突すると、サーバは QNAME が存在しないと証明する応答を返すことができない

    この場合、サーバは 2 (server failure) の RCODE の応答を返さなければならない(MUST)
  - このドキュメントで指定されたハッシュアルゴリズムである SHA-1 はそのような衝突はほとんどない

7. Authoritative Server Considerations (cont.)

- **7.3. Secondary Servers**
  - セカンダリサーバは、全てのhashed owner nameのNSEC3パラメータを見つけ出す必要がある。それは、複数のNSEC3 RRのセットから適切に選択できるようにするためである
  - これはzone apexにある NSEC3PARAM RR によって示される
  - NSEC3PARAM RRが複数存在するなら、有効な NSEC3 の連鎖が複数存在する
  - サーバはそれらを1つ選ばなければならないが、選択方法は問わない
- **7.4. Zones Using Unknown Hash Algorithms**
  - 本仕様に沿って署名されているが、未知のハッシュアルゴリズムを使っているゾーンは、正常に提供できない
  - そのような ゾーン は読み込み時に拒否するべきである(SHOULD)
  - そのようなゾーンに対するクエリに対してサーバ は RCODE=2(server failure) の応答を返すべきである(SHOULD)

7. Authoritative Server Considerations (cont.)

- **7.5. Dynamic Update**
  - NSEC3 を使用して署名されたゾーンは dynamic updates を使うことが出来る [RFC2136]

    ここではdynamic updates に関して、いくつか考察をおこなう

    ゾーンに名前の追加と削除を行うときは、empty non-terminal の作成もしくは削除をしなければならない(MUST)
    - ある名前とそれに対応するNSEC3 RRを削除するとき、その名前によって作成された empty non-terminals に対応する全ての NSEC3 RRs が削除されなければならない(MUST)

      削除される empty non-terminal は複数の名前に対応しているかもしれないので注意すること
    - ある名前とそれに付随するNSEC3 RRを追加するとき、全ての empty non-terminal に対応するNSEC3 RRを追加しなければならない(MUST)

      empty non-terminal にマッチする NSEC3 RR が存在していないなら、それを作成して追加しなければならない

7. Authoritative Server Considerations (cont.)

- Opt-Out を使用しているゾーンでは名前の追加または委譲でゾーン中のNSEC3 RR は必ずしも変更する必要はない
  - RRSet の 委譲を取り除くとき、その委譲がマッチするNSEC3 RRがないなら、それは opt out されているということである

    この場合、 さらに何かする必要はない。
  - RRSet の委譲を追加するとき、委譲のnext closer nameがOpt-Out NSEC3 RR によってカバーされているなら、その委譲はゾーンのNSEC3 RRの修正無しに追加してもよい(MAY)
- Opt Outを使用しているゾーンにNSEC3 RRを追加削除するとき、これから追加・変更しようとしているNSEC3 RRのOpt out flagの値は不明確である
- この不明確さをクリアにするために、サーバは以下のルールに従うべきである(SHOULD)
- これらのルールの主要な概念は、Dynamic UpdateをしようとしているゾーンをカバーしているNSEC3 RR のOpt-Out flag の状態が保持されているということにある。

7. Authoritative Server Considerations (cont.)

  - NSEC3 RR を削除するとき、チェインの１つ前の NSEC3 RR の Opt-Out flag の値は変更するべきではない。
  - NSEC3 RR を追加するとき、Opt-Out flag の値は追加しようとしているNSEC3 RRのowner nameを、今までカバーしていた NSEC3 RR の Opt-Out flag の値と同じにする
- もし、ここで述べているゾーンがOpt-Out flag の使用に関して一貫している、つまりゾーン中の全てのNSEC3 RRが同じフラグの値ならば、これらのルールは一貫性を持つだろう。
- もし、ゾーンが flag の使用で一貫性が無い、たとえば部分的にOpt-Outされているゾーンならば、これらのルールは Opt-Out flag の使用に関して、同じパターンをとれないだろう
- 部分的にOpt-Outされているゾーンに関して、理に適った方法があるならば、その方法はサーバのローカルポリシーに従って運用される

8. Validator Considerations

- **8.1. Responses with Unknown Hash Types**
  - validator は hash type が不明な NSEC3 RR を無視しなければならない。このような RR のみを含むレスポンスは通常 bogus として扱えばよい。
- **8.2. Verifying NSEC3 RRs**
  - validator は flag フィールドが 0 か 1 以外の値を持つ NSEC3 RR を無視しなければならない。
  - レスポンスに含まれる RR の以下のフィールドが互いに異なる場合は bogus 扱いとしてもよい:
    - hash アルゴリズム
    - iteration 値
    - salt 値
  - Section. 7.2. の要求に対する検証

8.3. Closest Encloser Proof

- **closest encloser proof を確認するためには、validator は以下の条件を満たす最長の名前 X を探す必要がある:**
  - X は応答に含まれる NSEC3 RR にマッチする QNAME の ancestor である。
  - X より1ラベル長い名前 (これも QNAME の ancestor、または QNAME そのもの) は応答に含まれる NSEC3 RR によってカバーされる

8.3. Closest Encloser Proof (cont.)

- **この proof を検証するアルゴリズムの例を示す:**
  1. QNAME を SNAME とおき、フラグをクリアする
  2. SNAME に対して
     - 応答中に SNAME にマッチする NSEC3 RR が存在しない場合、クリアする
     - 応答中に SNAME をカバーする NSEC3 RR が存在する場合、フラグを立てる
     - 応答中にマッチする NSEC3 RR が存在し、かつフラグが立っている場合、証明は完了。SNAME が closest encloser である
     - 応答中にマッチする NSEC3 RR が存在し、かつフラグが立っていない場合、応答は bogus である
  3. SNAME の左側からラベル1つ削除して 2. に戻る

8.3. Closest Encloser Proof (cont.)

- closest encloser が見つかった場合、validator は NSEC3 RR が closest encloser を持つこと、および original owner name が適切なゾーンに存在することを確認しなければならない。
- DNAME type bit はセットされてはならず、NS type bit は SOA type bit がセットされている場合に限りセットされ得る
- これにあてはまらない場合、攻撃者によってサーバに権威がない RR の存在が不正に否定されている可能性がある
- 以後、“X に対する closest (provable) encloser proof” とは、ここで述べた (あるいは等価な) アルゴリズムによって、X の ancestor が X の closest encloser であり、X が存在しないことを示すものとする

8.4. Validating Name Error Responses

- **Name Error Response を受け取った場合、validator はレスポンスに以下が含まれていることを確認しなければならない:**
  - QNAME に対する closest encloser proof
  - closest encloser のワイルドカードをカバーする NSEC3 RR
- **Section 7.2.2. の要求に対する検証**

8.5. Validating No Data Responses, QTYPE is not DS

- **No Data Response を受け取った場合、validator はレスポンスに含まれている NSEC3 RR に対して以下を確認しなければならない:**
  - QNAME が一致する
  - Type Bit Maps に QTYPE で指定した TYPE または CNAME が含まれていない
- **Section 7.2.3. の要求に対する検証**

8.6. Validating No Data Responses, QTYPE is DS

- **QTYPE が DS のクエリに対して No Data Response を受け取った場合:**
  - そのような NSEC3 RR が存在しなければ、以下を確認しなければならない:
    - QNAME に対する closest provable encloser proof がレスポンスに含まれる。
    - “next closer name” をカバーする NSEC3 RR の Opt-Out bit がセットされている。
- **Section 7.2.4. の要求に対する検証**

8.7. Validating Wildcard No Data Responses

- **ワイルドカードを使ったクエリに対して、 No Data Response を受け取った場合、validator は以下を確認しなければならない:**
  - QNAME に対する closest encloser proof
  - レスポンス中に “*.<closest encloser>” にマッチする NSEC3 RR が存在すること
  - この NSEC3 RR にクエリ中で QTYPE として指定した TYPE および CNAME がセットされていないこと
- **Section 7.2.5. の要求に対する検証**

8.8. Validating Wildcard Answer Responses

- ワイルドカードを使ったクエリに対する検証された応答が含む RRSet は、validator に QNAME に対する closest encloser (の候補) を提供する。
- この closest encloser はワイルドカードの直近の ancestor である
- validator は、QNAME に対する “next closer” name をカバーする NSEC3 RRがレスポンス中に存在することを確認しなければならない
- これにより、QNAME そのものは存在せず、レスポンスの生成にはワイルドカードが使われたことが示される
- Section 7.2.6. の要求に対する検証

8.9. Validating Referrals to Unsigned Subzones

- **応答に委譲名にマッチする NSEC3 RR が含まれる場合、validatorは以下を確認しなければならない:**
  - Type Bit Maps に NS が含まれ DS が含まれない
  - これは Type Bit Maps に SOA が含まれない – NSEC3 RR が適切なゾーンに所属しているかの確認
- **NS bit が立っている場合、DNAME bit が立っていないことを示すので、NSEC3 RR 中の Type Bit Maps の DNAME bit の確認をおこなう必要はない**
- **委譲名にマッチする NSEC3 RRが存在しなかった場合、validatorは以下を確認しなければならない:**
  - 委譲名に対する closest provable encloser proof
  - 委譲名への”next closer” name をカバーする NSEC3 RR のOpt-Out bitがセットされていること
- **Section 7.2.7. の要求に対する検証**

9. Resolver Considerations

- **9.1. NSEC3 Resource Record Caching**
  - キャッシュを利用するリゾルバは、応答が NSEC3 RR を必要とするような場合に、適切な NSEC3 RR を取得できなければならない。
    - NSEC の場合は応答に必要とされる NSEC RR を名前を用いて見つけることが可能だが、NSEC3 では応答に必要な NSEC3 RR の名前を計算するのは困難である。
    - リゾルバの実装は NSEC3 RR をキャッシュ、および取得するための新たな手段が必要になるかもしれない。

9.2. Use of the AD Bit

- **“next closer” name をカバーする NSEC3 RR の Opt-Out bit がセットされているような closest encloser proof を含むレスポンスを返す場合、AD bit をセットしてはならない。**
- **この closest encloser proof は何を証明したいのか:**
  - Opt-Out bit がセットされた NSEC3 RR によってカバーされる名前は insecure かもしれないし、そうでないかもしれない。
  - このような場合の closest encloser proof を含むレスポンスに含まれている全てのデータが検証されているわけではなく、AD bit はセットされ得ない。

10. Special Considerations

- **10.1. Domain Name Length Restrictions**
  - ゾーン名は hashed owner name を含めても 255 文字以下になる長さでなければならない。
  - この制約の実際の値は使用するハッシュアルゴリズムに依存する。
  - 例)
    - SHA-1 が生成するハッシュ値は160bit、base32エンコードを適用すると32文字、’.’ を含めると 255 - 32 – 1 = 222 文字がゾーン名の最大長となる。

10.2. DNAME at the Zone Apex

- **RFC2672 で定義されている no-descendants 制約の更新**
  - ノードNにDNAME RRが存在する場合、Nの派生レコードにいかなるデータも存在してはならない。
- **NSEC3 および RRSIG RR は DNAME の有無に関わらず、zone apex の派生レコードとしての存在してよいものとする。**

10.3. Iterations

- Iteration 数はゾーンの署名、提供、検証の際に必要な計算コストに大きな影響を及ぼすので、この回数に制約を設ける。
- 最小の ZSK のサイズを表に値に従って切り上げ、対応する Iteration 回数を最大値とする。
- リゾルバはこの値を超える Iteration 数がレスポンスに含まれていたら insecure 扱いとしてよい。
- この閾値は鍵アルゴリズムに依らない。

鍵サイズ	Iteration 数
1024	150
2048	500
4096	2500

10.4. Transitioning a Signed Zone of from NSEC to NSEC3

- **既に (NSEC を使って) 署名されているゾーンを NSEC3 に移行する際の注意点**

1. **全ての DNSKEY RR を Section 2.で定義したアルゴリズムエイリアスで置き換える**
   - NSEC3 未対応クライアントがゾーンを insecure 扱いするようになる
   - ネガティブ応答およびワイルドカード応答はNSEC RRを含んだまま
2. **NSEC3 RR を徐々に、あるいは一度に追加する**
   - 徐々に追加する場合、NSEC3PARAM RR を最後に追加しなければならない
3. **NSEC3PARAM RRを追加する**
   - ネガティブ応答、およびワイルドカード応答にも NSEC3 RR が使われるようになる
4. **NSEC RR を徐々に、あるいは一度に削除する**

10.5. Transitioning a Signed Zone of from NSEC3 to NSEC

- **NSEC で署名されている状態に戻すには、10.4.で説明した手順を逆におこなえばよい**

1. **NSEC RR を徐々に、あるいは一度に追加する**
2. **NSEC3PARAM RRを削除する。**
   - ネガティブ応答、およびワイルドカード応答に NSEC RR が使われるようになる
3. **NSEC3 RR を徐々に、あるいは一度に削除する**
4. **全ての DNSKEY RR のアルゴリズム識別し戻す**
   - NSEC3 未対応クライアントがゾーンを secure 扱いするようになる

12. Security Conisderations

12.1. Hashing Conisderations

12.1.1. Dictionary Attacks

- **dictionary attack**
  - 取得した NSEC3 RR と適当な名前のハッシュ値とを比較してゾーン内の名前を列挙する
- **pre-calculated dictionary attack**
  - 事前に適当な名前のハッシュ値を計算しておき、定期的に NSEC3 RR をスキャンして比較することによるゾーン内の名前の列挙
  - 定期的な salt の変更によって防ぐ
- **salt 値は 64 bit 以上の長さを持ち、予測不可能な値にするべきである**

12.1. Hashing Conisderations (cont.)

12.1.2. Collisions

- **QNAME と NSEC3 RR の owner name のハッシュ値の衝突**
  - 存在しないとは言い切れないが、ほぼ発生し得ないレベル
  - DNSSEC は既に第2原像計算困難性に依存している (ので今更ハッシュ値の衝突の議論なんかするな)
    - 第2原像計算困難性：ある入力 m1 があるとき、hash(m1) = hash(m2) となるようなもう1つの入力 m2（m1とは異なる入力）を見つけることが困難である

12.1.3. Transitioning to a New Hash Algorithm

- **NSEC3 および NSEC3PARAM RR にはハッシュアルゴリズムを指定するフィールドがある**
- **しかし、ハッシュアルゴリズムを安全に切り替える方法、別途定める必要がある**
- **実際には、切り替えには、NSEC RR を使用する状態、または insecure な状態のいずれかを経由する必要があるかもしれない**

12.1. Hashing Conisderations (cont.)

12.1.4. Using High Iteration Values

- **validator は大きな iteration 値を持つ NSEC3 RR を含む応答を insecure 扱いすべきとされているため、署名された大きな iteration 値を持つ NSEC3 RR は downgrade 攻撃を可能にしてしまう**
- **攻撃者は応答から任意の NSEC3 RR を削除し、大きな iteration 値を持つ NSEC3 RR で置換、あるいは追加する。validator は応答を insecure 扱いすることとなる**
- **この攻撃は以下の全ての条件を満たす場合に成り立つ:**
  - ゾーン内に大きな iteration 値を持つNSEC3 RRが1つ以上存在する
  - 攻撃者がこのような NSEC3 RR のうち1つ以上にアクセス可能である。これは、このような NSEC3 RR が典型的な応答に含まれる場合に成り立つ。また、AXFR, IXFR またはその他の方法で攻撃者がゾーンにアクセスできる場合にも成り立つ
- **大きな iteration 値はサーバへの DoS の可能性も高める**
  - サーバはネガティブ応答やワイルドカード応答のためにハッシュ値の計算をおこなう必要があるため

12.2. Opt-Out Considerations

- **Opt-Out フラグによって署名されたゾーンの中に署名なしの名前を作成することができる**
- **署名なしの名前は insecure であり、その正当性や存在は暗号技術によって証明されることはない**
- **一般に**
  - 署名なしの名前を持つ RR は署名なしゾーン内の RR と同様の脆弱性を抱える
  - 署名された名前を持つ RR は Opt-Out フラグの有無に関わらずセキュリティ条件は等しい
- **Opt-Out の有無に依らず、insecure delegation は攻撃者に改変される可能性がある**
- **セキュリティ面から見た Opt-Out の最大の特徴は insecure delegation の存在/非存在証明ができなくなる点**

12.2. Opt-Out Considerations (cont.)

- **つまり、悪意のあるエントリによって、署名なし RR の挿入や削除をおこなうことができる**
  - 通常 NS RR を用いるが、署名付きワイルドカードの展開も用いられる
  - ワイルドカード RR には署名がされているが、展開された名前は署名なし
- **委譲を設定できることは、任意の RR を追加できることと等価である**
  - 攻撃者は、攻撃者が制御できる別のネームサーバへの委譲を設定しさえすればよい
- **この影響は軽んじられるべきではなく、Opt-Out フラグは慎重に用いられなければならない**
- **特に、ゾーンへの署名をおこなうツールはデフォルトで Opt-Out が有効になるべきではなく、Opt-Out を全くサポートしない、という選択をしてもよい**

12.3. Other Considerations

- NSEC3 RR を全て取得することで、ゾーン中の RR の数およびタイプは公開されることになる
- ダミーのエントリを追加することで、それらの情報は厳密ではなくなるが、それでも上限値は公開することになる