安心の鍵は変わらぬ愛と親子の絆
～ランチのおともにDNS～

2010年11月25日

Internet Week 2010 ランチセミナー

株式会社日本レジストリサービス

森下泰宏（オレンジ）・民田雅人（みんみん）

本日のお題

1. いよいよ始まったDNSSEC
   - これまでの道程と努力を簡単に振り返る
2. 運用者からみたDNSSECと今までのDNSの違い
   - DNSSECはこれまでのDNSと何が違うのか?
3. 実際の障害例から学ぶDNSSECの運用ポイント
   - 何がいけなかったのか? どうすればいいのか?
4. 大切な「変わらぬ愛」と「親子の絆」
   - 愛と絆が全体を支える

いよいよ始まったDNSSEC

- 20年にわたる「長く曲がりくねった道程」（のごく一部）

1990年 DNSセキュリティに関する最初の論文
1993年 IETFにおいて最初のDNSSEC BOF開催
1997年 DNSSECの最初のRFC（RFC 2065）発行
1999年 RFC 2535発行、BIND 9プロジェクト開始
2000年 IETF DNSEXT WG活動開始
2003年 DS（RFC 3658）発行
2005年 DNSSECbis（RFC 4033, 4034, 4035）発行
2007年 トラストアンカー自動更新（RFC 5011）発行
2008年 NSEC3（RFC 5155）発行
2008年 カミンスキー型攻撃手法の発表
2010年 ルートゾーンの署名開始
2011年 JP DNSSEC**正式サービス開始**

これまでのさまざまな努力により…

- できるだけ運用の手間がかからないように
- 従来のDNSの手法に近い形で管理できるように
- 従来の登録システムに沿った形で管理できるように
- 大規模なTLDにも段階的に導入できるように
- 既存のものへの影響を最小限に留めるように…

…DNSSECの導入が進められてきた

- しかし、どうしても吸収できなかった「違い」がある

⇒ その「違い」とは?

DNS運用者の目から見た「違い」とは?

DNSSECは今までのDNSと何が違うのか?

これまでのDNSとの違い

- それぞれの要素における「違い」
  - DNSSECと「時間」
  - DNSSECと「データ」
  - DNSSECと「タイミング」
  - DNSSECと「順番」

DNSSECと「時間」

- これまでのDNSに存在する時間は全て**相対時間**
- データが見つかった時点で有効
- データを**受け取ってから**タイマーが発動
  - TTL
  - ネガティブキャッシュ
  - SOAの各種パラメータ、など
- タイマーはデータを取り扱う各要素（権威DNSサーバー、キャッシュDNSサーバーなど）が**個別に管理**
  - 相対時間であるため、それぞれのサーバー間での**時刻の同期は必要なかった**
- SOAのシリアル番号は**時間ではない**

DNSSECは絶対時間を要求する

- DNSSECの署名（RRSIGレコード）には、有効期間の**始まりと終わりの双方**が、1970年1月1日からの**絶対値**で入る
  - Signature Inception（署名の有効期間**開始**）
  - Signature Expiration（署名の有効期間**終了**）
- 署名する側
  - **自身の時計**を参照し「この署名はいつからいつまで有効にする」ということを意識して署名
- 検証する側
  - **自身の時計**を参照し「受け取った署名が有効期間の範囲内かどうか」をチェック

DNSSECは系全体での時刻同期が必須

- 署名が**あっても無効**、という状態が起こりうる
- **系全体で時間を合わせておく**必要あり
- 時間が大幅にずれると、**署名検証に失敗**する
- みなさんは時間をどうやって合わせていますか?
  - /etc/ntp.confがDNSを参照していたりしませんか?

DNSSECと「データ」

- これまでのDNSでは、同じものに由来する権威あるデータを親子双方で持つことはなかった
  - 親が持つNSやグルーAは**権威あるデータではなく**、あくまでも参考情報に過ぎない
  - NSやAは、**子のもののみ**が権威あるデータ

同じものに由来する権威あるデータを親子双方が保持

- DNSSECでは、子のDNSKEY（KSK）から生成した**DS**を親に登録する
- DSの導入により、NSの**ような**管理が可能になった
  - それまでは「子の鍵に親が署名して子に戻す」が必要
- 親のDSと子のDNSKEY（KSK）は、**双方とも権威あるデータ**となる
  - 従来のNSやAとは異なる

DSとDNSKEY（KSK）の密接な関係

- 親がDSを持っている場合、子は対応するDNSKEY（KSK）を**必ず持っていなければならない**
  - 親のDSに対応するDNSKEY（KSK）が子に存在しない場合、その時点で**DNSSEC検証エラー**になる
- 親にDSがあるが子に対応するDNSKEY（KSK）がない、という状態は**あってはならない**
  - DSやKSKを操作する場合、「**タイミング**」や「**順番**」が重要
  - DSやDNSKEYが持つTTL値にも注意が必要（後述）

DNSSECと「タイミング」

- DNSSECではデータが**存在しても（まだ・もう）有効ではない**、という状況が発生する
  - 有効期間外の署名
  - これから使う鍵
- 管理運用（例: ヘルスチェック）の際には、**単なるデータの存在チェックだけでは不十分**
  - 有効期間（いつからいつまでの間）のチェックが必要
  - チェックする側も時刻合わせが必要

DNSSECと「順番」

- データの更新（特に親子に関係するもの）の**順番**に注意する必要がある
- 子の準備完了 → **インターネット全体への伝搬** → 親のデータを登録（更新）という手順を、これまで以上に遵守する必要がある
- 典型的な例
  - あるドメイン名に対するDNSSECの新規導入
  - あるドメイン名に対するDNSSECの中止

これまで以上にTTLへの意識が必要

- 設定変更の際、TTL満了のタイミングを**これまで以上に意識する**必要がある
  - 自分のゾーンに設定したTTL
  - **親が設定しているTTL**（DSレコード）
- 親がDSレコードに設定するTTLは**親ごとに異なる**
- 典型的な例
  - KSKの更新（ロールオーバー）

例1: KSKのロールオーバー

1. 新しいKSKの生成と自分のゾーンデータへの追加
2. 新旧双方のKSKでDNSKEY(ZSK)を署名
3. 自分のDNSKEYの伝搬を確認
   - **自分のDNSKEYのTTL時間分の経過を待つ**
4. 新しいKSKに対応するDSレコードを親に登録申請
5. 親のDSの登録更新を確認
6. 親のDS登録更新の伝搬を確認
   - **親のDSのTTL時間分の経過を待つ**
7. 古いKSKをゾーンデータから削除
8. 新しいKSKのみでDNSKEY(ZSK)を署名

例2: DNSSECの新規導入

1. 新しいZSK、KSKの生成とゾーンデータへの追加
   - 事前公開するZSKも忘れずに生成しておく
2. ZSKによるゾーンデータへの署名と公開
3. KSKによるDNSKEY（ZSK）への署名と公開
4. KSKに対応するDSレコードを親に登録申請
5. 親のDSの登録を確認
6. DNSSEC検証ありで名前が引けることを確認

例3: DNSSECの中止

1. DSの削除を親に申請
2. 親のDS登録の削除を確認
   - **親のDSのTTL時間分の経過を待つ**
3. 自分のDNSKEYの削除・署名データ、NSEC/NSEC3関連データの削除

実際の障害例から学ぶ DNSSECの運用ポイント

本日紹介する4つの障害事例

- RIPE NCC
  - 欧州地区逆引き、e164.arpaなど
- Nominet UK
  - .ukのレジストリ
- mozilla.org
  - Mozilla Project
- iab.org
  - 「インターネットの技術コミュニティ全体の方向性やインターネット全体のアーキテクチャについての議論を行う技術者の集団」（http://www.nic.ad.jp/ja/basics/terms/iab.html より）

障害例1: RIPE NCC

- 障害発生日: 2010年9月21日（中央ヨーロッパ時間）
- 障害の範囲
  - RIPE NCCが管理するすべての署名済みゾーン
- 起こったこと
  - DNSSEC署名の有効期間の開始と終了の双方が0（1970年1月1日）に設定・公開された
- 原因
  - KSKのロールオーバーにおけるレジストリシステム上の不具合

    [dns-wg] Postponement of DNSSEC KSK Roll-over for RIPE NCC Zones
    <http://www.ripe.net/ripe/maillists/archives/dns-wg/2010/msg00076.html>
- 対応方法
  - 緊急にひとつ前のデータにゾーンをロールバックし、対応完了後に正規のデータに更新

障害例2: Nominet UK

- 障害発生日: 2010年9月11日（英国時間）
- 障害の範囲
  - .uk
- 起こったこと
  - ZSKと署名の間に矛盾が発生
- 原因
  - DNSSEC関連ハードウェア（HSM）に障害が発生しバックアップシステムに切り替えた際、バックアップ側で運用中のものと同じZSKを使用していなかったため、ZSKと署名の間に矛盾が発生した

    DNSSEC incident report <http://blog.nominet.org.uk/tech/wp-content/uploads/2010/09/dnssec-incident-report.pdf>
- 対応方法
  - キャッシュDNSサーバーの再起動（キャッシュクリア）を促した

障害例3: mozilla.org

- 障害発生日: 2010年9月16日（米国時間）
- 障害の範囲
  - mozilla.org
- 起こったこと
  - mozilla.orgのDNSSEC検証が失敗する状態になった
- 原因
  - mozilla.orgへのDNSSEC導入の際、親へのDSの登録・公開をmozilla.org自身の鍵の公開・署名よりも先に実施してしまった
- 対応方法
  - mozilla.org自身のDNSSEC鍵の公開・署名を実施

障害例4: iab.org

- 障害発生日: 2010年8月31日（米国時間）
  - DNSSEC is hard to get right
    http://www.ietf.org/mail-archive/web/ietf/current/msg63269.html
- 障害の範囲
  - iab.org
- 起こったこと
  - iab.orgの署名期限切れ
- 原因
  - 人為的ミス(うっかり?)により、8月31日の時点で署名の有効期間終了が「20100829223019（2010年8月29日22時30分19秒）」になっていた
  - 外部からの報告により発覚
- 対応方法
  - 再署名の実施
  - 再発防止策については未アナウンス（だと思う）

鍵と署名に関連する障害を防ぐために

- 可能なものは自動化する
  - BIND 9.7のスマート署名や全自動ゾーン署名
  - OpenDNSSECなどの専用ツールの導入
- よりよいヘルスチェックの実施
  - 運用経験の蓄積と障害の防止
    - ドキュメントの整備
    - ノウハウの共有
  - 障害発生時の復旧手順の確立
    - 有効期限外の署名は実は復旧が早い（キャッシュされない）
    - 鍵と署名の矛盾、DSとDNSKEY（KSK）の矛盾が起こると致命的

大切な「変わらぬ愛」と「親子の絆」

変わらぬ愛

- 従来は、愛し続けなくてもDNSはそれなりに動いた
  - 最初は愛情こめてきちんと設定
  - しばらくすると愛も薄れ、設定しっぱなし（意識の外になりがち）

- DNSSECでは、DNSへの「**変わらぬ愛**」が必要
  - **定期的な**再署名
  - **定期的な**鍵（ZSK、KSK）の更新

親子の絆

- 従来は、いい加減(?)なつながりでもDNSはそれなりに動いていた
  - 親のNS・グルーAと子のNS・Aが一致していない
  - 複数サーバのうち1台がLame Delegationの状態
  - etc...

- DNSSECでより大切になる「親子の絆」
  - DSとDNSKEY（KSK）が作る、**より緊密な親子関係**
  - TTLにまで気を配った、**慎重な親子関係の構築**

まとめ: 愛と絆が全体を支える

- DNSSECではDNSが**本来持つべき**「変わらぬ愛」と、「親子の絆」が、これまでよりも**より大切になる**
- それぞれが管理するDNSに対する「愛」と「絆」が、**インターネット全体のDNSを支える力**となる

- 2011年1月16日、JPドメイン名でのDNSSEC正式サービス開始

皆様、どうぞよろしくお願いいたします