RFC 4986
プロトコル理解SWG

DNSSEC.jp

タイトル

Requirements Related to DNS Security (DNSSEC) Trust Anchor Rollover

DNSSECトラストアンカーロールオーバーに関する要件

Abstract

- すべてのsecurity-awareなリゾルバは、最低1つのトラストアンカーが必要である。
- さまざまな事情により、数個のトラストアンカーを設定できることが求められている。
- トラストアンカーの監視と更新を手動で行うことができる運用環境もあるが、多くの場合、security-awareなリゾルバによる自動的な更新が必要である。
- このドキュメントは、security-awareなリゾルバが自動的なトラストアンカーのロールオーバを実装する際に満たさなければならない要件を明確にするものである。

1. Introduction

- DNSSECは、security-awareなリゾルバに保持されている1つ以上のトラストアンカーからDNSリソースレコードを検証できるように新しいレコードとプロトコルの変更を定義している。
- まず最初に、正しく有効であることを確実に保証してくれる信頼できる経路でトラストアンカーを取得しなければならない。
- 最初のトラストアンカーの取得方法は複数あるが、この文章のスコープ外である。
- 一度運用者がトラストアンカーを取得したら、手動でリゾルバにトラストアンカーを初期設定をすることになるだろう。

1． Introduction

- 手動でトラストアンカーの管理を行える運用環境もあるだろう。
- しかしながら、多くの運用環境では、トラストアンカーの更新と管理を明確な方式でより自動化されることを求めているだろう。
- このドキュメントは、今後提案(IETFに対して、もしくは独自として※)される自動的なトラストアンカーのロールオーバー手法に対して、その有効性を確認するために必要な要件リストを定義する。

※RFCに書いてないが範囲を限定していないという事を言いたい。

2. Terminology （用語）

- “MUST“, ”MUST NOT“, ”REQUIRED“, ”SHALL“, ”SHALL NOT“,”SHOULD“, ”SHOULD NOT“, ”RECOMMENDED“, ”MAY“, and ”OPTIONAL“ は、RFC２１１９を参照
- 要件に競合が起こった場合、MUSTは、SHOULD, MAY, RECOMMENDEDより優先される。

3. Background

- DNSの回答を得るのには、1つ以上の参照点が必要である。
- スタブ・リゾルバの参照点は、一般に1つ以上の再帰的なネームサーバのIPアドレスである。
  (/etc/resolv.conf)
- 再帰的なネームサーバの参照点は、一般的にはrootネームサーバのIPアドレスである。
  (/etc/bind/db.root　ubuntu10.04)
- security-awareなリゾルバも、署名されたDNS応答を検証するための認証チェインとして1つ以上の参照点を持たなければならない。

3. Background

- DNSSECでは、IPアドレスに代わる参照点として、信頼できる1つ以上のDNSKEY もしくはDS リソースレコードが必要となる。これらの参照点をトラストアンカーと呼ぶ。
- トラストアンカーは、いつ「トラストアンカー」と呼べるようになるのか。DNSKEY とDS リソースレコードは、署名を行うゾーンの運用者によって作られた時点でトラストアンカーになるのではないし、署名されたゾーンで公開された時点でトラストアンカーになるのではない。
- security-awareなリゾルバの運用者がその公開鍵もしくはハッシュをトラストアンカーとして使うと決定した時点で、トラストアンカーとなる。

3. Background

- リソースレコードを作成および／または公開しているゾーンの運用者は、自分DNSKEYもしくはDSリソースレコードがsecurity-awareなリゾルバからトラストアンカーとして使われていることを知らないかもしれない。
- 一方、RootゾーンのDNSKEYリソースレコードは、多くのsecurity-awareなリゾルバからトラストアンカーとして利用されているという点で明らかな違いがある。
- さまざな理由によるが、root以外のゾーンのDNSKEYもしくはDSリソースレコードはsecurity-awareなリゾルバの管理者によってトラストアンカーとして指定されうるものである。

3. Background

- 署名されたDNS応答を検証するための認証の連鎖を構築し、開始点としてsecurity-awareなリゾルバがトラストアンカーを使う際にはsecurity-awareなリゾルバの運用者が自らが選んだDNSKEY もしくはDSリソースレーコードが有効であることを確保する責任を負っている➡

  自由に設定できるので責任は使う側

- security-awareなリゾルバの運用者が1つ以上のトラストアンカーを選ぶ際には、有効なリソースレコードを利用し続けられること、トラストアンカーとして利用しているリソースレコードが変更になったり削除された場合は対応する手段を決めておかなければならない。

3. Background

- 黎明期でDNSゾーンに署名をした人々は、security-awareなリゾルバの運用者が適切にトラストアンカーを手動で変更することができるように、管理してるゾーンのDNSKEYやDSリソースレコードを変更する際の過程や方式を公開している。
- この手動によるアプローチは、おそらく広がらないだろう。
- それゆえに、ロールオーバを自動で行う取り組みが必要とされている。

4. Definitions

- 本RFCでは、RFC4033の２章（重要なDNSSEC用語の定義）に加えて、以下の定義を用いる。

4. Definitions

Trust Anchor:

- DNSKEYもしくはDNSKEYのハッシュとして定義されているDSリソースレコード（RFC4033参照）のこと。
- security-awareなリゾルバはこの公開鍵もしくはハッシュを認証を辿っていく際の参照点として利用する。加えて、このDNSKEYもしくはDSリソースレコードはDNS階層のなかの1つのポイントに正確に関連付けられている。すなわち、1つのDNSゾーンとなっている。
- 複数のトラストアンカーが、それぞれのDNSゾーンに関連付けられていて、さまざまなリゾルバによって設定されていてもよい。リゾルバは、複数のDNSゾーンの中からトラストアンカーを設定してもよい。
- リゾルバの運用者は、利用するトラストアンカーのリソースレコードの組み合わせを選んでいるので責任を負っている。

4. Definitions

Initial Trust Relationship:

- 最初に信頼できる方法でトラストアンカーを入手しなければならない。
- 例えば　ルートゾーンのDNSKEYリソースレコードを得る際は
  - IANAのWebsite
  - ルートゾーンとして公開されているDNS情報
  - よく知られたハードコピーの中に公開鍵が公布されているもの

  などの複数の情報源から比較することで検証する。
- その他のトラストアンカーは、設定する前に正しく有効であるものを確保しなければならない。技術的、手続的、契約的な関係の組み合わせかもしくは　現在のDNSプロトコルの外の実在の信頼関係の組み合わせを使って達成されるであろう。

4. Definitions

Trust Anchor Distribution:

- 署名されたゾーンとsecurity-awareなリゾルバの運用者の間でDNSKEY もしくはDSリソースレコードを伝えるための手順もしくは手順群。
- これらの手順もしくは手順群は、security-awareなリゾルバの運用が新しいRRをTrust Anchorとして設定する際に、Initial Trust Relationshipを維持できている認証性と健全性を確保できて、かつ十分信用できる手法と認められるものでなくてはならない。[MUST]

4. Definitions

Trust Anchor Maintenance:

- 検証を行っているリゾルバに対して、新しいトラストアンカーを追加したり、既存のトラストアンカーを削除したり、別のものに置き換えたりする変更のこと。
- これらの変更は、手動もしくはいくつかの自動化された方法で達成されるだろう。
- security-awareなリゾルバの運用者は、まずInitial TrustRelationshipを導入する手順と方針を決めておかなければならない。それができたら、トラストアンカーを設定してよい。

4. Definitions

Trust Anchor Revocation and Removal :

- トラストアンカーとなっているDNSKEYやDSリソースレコードを署名を行ったゾーンの運用者が取り消したり除去したりする際に行われる特定のトラストアンカーの無効化処理のこと。
- ゾーンの管理者が望む時に1つのポイントで1つ以上のリソースレコードを無効化することが可能である。
- ゾーンの管理者がDNSKEYかDSを消した場合（かつそれがトラストアンカーとして使われていた場合）、「どれ」が消されたか、という情報をトラストアンカーを設定している側に伝えないといけない。DNSKEYやDSが複数ある場合の対策。
  [MUST]

（レコードとトラストアンカー自体の削除のこと）

4. Definitions

Trust Anchor Rollover:

- security-awareなリゾルバによって保持されている1つもしくは複数のトラストアンカーを安全に置き換えるために必要な方法や方法群。
- Trust Anchor Rolloverは、Trust Anchor Maintenanceのサブセットとして考慮されているものでなくてはならない。

Normal or Pre-Scheduled Trust Anchor Rollover:

- DNSSECで署名されたゾーンの運用者が、定常業務として新しいDNSKEYおよび／またはDSリソースレコードを発行するケース。

4. Definitions

Emergency or Non-Scheduled Trust Anchor Rollover:

- 署名されたゾーンの運用者が、新しいDNSKEYおよび／またはDSリソースレコードを異例な状況で発行するケース

Emergency Trust Anchor Revocation:

- 署名されたゾーンの運用者が、現状のDNSKEYおよび／またはDSリソースレコードを異例な状況で失効させるケース

5. Requirements

以下は、自動化されたトラストアンカーリゾルバのロールオーバを実現するための要求仕様となっている。

5.1 Scalability

- インターネット規模の利用に対してスケーラブルでなければならない。[MUST]
- トラストアンカーのロールオーバーが必要になるケースで最大規模のものは、おそらくトラストアンカーとして利用されているルートゾーンの公開鍵になるだろう。
- この数は、組み込みとして出荷されたsecurity-awareなリゾルバの数によっては極端に大きくなることがありうる。

- 自動化されたトラスアンカーのロールオーバの実装は、それぞれのDNSゾーン単位で複数のゾーンとトラストアンカーをサポートできなければならない。[MUST]
- security-awareなリゾルバに設定できるトラストアンカーの数を今回は明確にはしていないが、多くのケースで20以下だが、場合によっては1000程度になるだろう。

5.2 No Known Intellectual Property Encumbrance

既知の知的財産権の負担なし

- トラストアンカーのロールオーバーは必要不可欠なものであるから、実装で選ばれるものは8章の[5] “Intellectual Property Rights in IETF Technology”,RFC 3979の負担（権利を主張されいるもの）であるものを選んではならないのか、もしくは、ロイアリティフリー（特許技術などを利用し製品を作成する際にライセンスを取得する必要のないもの）でなければならない。

5.2 No Known Intellectual Property Encumbrance

- この目的のために、ロイアリティフリーは、以下のように定義する。
- レジストリ・レジストラ・権威/再帰的/キャッシュ/転送/スタブリゾルバなどそれに類するなどすべてのドメインネーム　サービスのアプリケーションの範囲で
- 特定の国で制限されない
- 商用かそうでないも含めて合わせて利用料金なし
- 今後改変できない権利を有するもの
- アルゴリズムの記述を利用に際して含むもの
- ハードウェアでの利用、ソースもしくはバイナリ形式でのソフトウェアの利用が可能なもの

5.2 No Known Intellectual Property Encumbrance

- 要約すると、トラストアンカーroll overのの技術を実装、配布、運用する者は、いかなるIPRホルダー(知的財産権所有者)に対して、何ら費用を支払う必要が無い状況が望まれる。

5.3 General Applicability（一般的な利用可能性）

- 実現方式は、どのDNSのゾーン（すべてを含み）を対象にしたとしてもリゾルバーに設定されたトラストアンカーを維持できるものでなければならない。[MUST]

5.4 Support Private Networks

- 実現方式は、プライベート環境での独自のDNS階層でも利用できなければならない[MUST]

5.5 Detection of Stale Trust Anchors

失効したトラストアンカーの検出

- 実現方式では、リゾルバ側で既存で設定しているDNSKEYもしくはDSリソースレコードを用いてトラストアンカー更新ができなくなってしまった事象が発生した場合、リゾルバ側でそれを検知できる必要がある。[MUST]
- この場合、リゾルバ側からゾーン管理者に対して、Initial trust relationshipを構築しなおす事を提案するべきである。

5.6 Manual Operations Permitted

- security-awareなリゾルバの運用者は、手動もしくは自動のロールオーバを選択してもよい。
- しかし、ロールオーバプロトコルは自動と手動の両方のトラストアンカーの保守操作を行えるように実装しなければならない。
- ロールオーバプロトコルの実装は、ほぼ必須要件であるが、この要求仕様の範囲外である。

5.7 Planned and Unplanned Rollovers

- 実装方式は、計画された（事前に予告されたもの）と計画されていない（予告なし）のトラストアンカーロールオーバが行えるようになっていなければならない（MUST）
- Initial Trust Relationshipをサポートすることはオプションである。

5.8 Timeliness（時を得たこと）

- トラストアンカーとして用いられるリソースレコードは、折よい方法でsecurity-awareなリゾルバに伝搬できるようになっているべきである。（SHOULD）
- 現行のDNSKEYもしくはDSリソースレコードを更新（追加・取り消し）を行った場合に長時間トラストアンカーとして使われ続けないように実装する必要がある。

5.9 High Availability （高可用性）

- ゾーン管理者は、トラストアンカーとして用いられるリソースレコードの状態(有効、無効など)について、常に信頼できる方法でリゾルバに対して情報公開している事が望ましい。[SHOULD]

- ゾーンの管理者が無効化した際に、トラストアンカーとして用いられていることが分かっているリソースレコードに関する情報は、十分な期間、信頼できる方法で入手できるようにするべきである。

5.10 New RR Types

- 新しいリソースレコードのタイプかプロトコルの変更を必要とする実現方式が提案された場合、その方式を評価する過程で考慮されるべきである。
- ワーキンググループがその変更が良いものか、悪いものか　それとも別の方法があるかを決める。

5.11 Support for Trust Anchor Maintenance Operations

トラストアンカーの維持業務のためのサポート

- トラストアンカーロールオーバの実現方式は、検証を行っているsecurity-awareなリゾルバに対して新しいトラストアンカーを加えたり、既存のトラストアンカーを削除したり、もしくは他のものに置き換えたりする操作ができるようになっていなければならない。[MUST]

5.12　Recovery from Compromise(漏洩からの回復)

- トラストアンカーロールオーバーの実装方式は、いくかの鍵の漏洩に際して、そのゾーンとして設定されている漏洩されていない最低1つの他の鍵が残っている限り、信用を回復できるようになっていなければならない。[MUST]

5.13　Non-Degrading Trust(デグレードなしの義務)

- ロールオーバを実行するのに際して既存の信頼関係がデグレードしないように信憑性と完全性を確保するのに十分な手段を確保しなければならない。[MUST]

6.　Security Considerations （セキュリティの考慮）

- このドキュメントは、security-awareなリゾルバが自動化されたトラストアンカーロールオーバの実装を行うための総合的な要求仕様を定義しているものではあるが、これらの要求に必要なセキュリティの仕組みについては定義していない。

7.　Acknowledgements （承認）

- このドキュメントは、DNSSEC　トラストアンカーリゾルバーに関する要求の話題で　DNSEXTワーキンググループのメンバーの大多数の意見を反映したものである。
- ワーキンググループの多くのメンバーの貢献によって読みやすく改善され、このドキュメントの形式は、ありがたくも一般に承認されたものとなった。

8.　Normative References

- 参照　　以下省略

以上