Vol.2013-IOT-23 No.5
2013/9/27

失効したルート鍵を持つ DNSSEC 検証サーバの挙動調査と対策についての一考察

石原　知洋[†]　　櫨山　寛章[††]　　関谷　勇司[†]

DNSSEC は DNS のデータの完全性を確保するための仕組みであり、ルート含むドメインツリーの特定の点（トラストアンカー）からその下位ドメインの鍵を連鎖的に署名することで、データの正当性を保証する。このとき、正当性の検証のために、トラストアンカーの公開鍵を DNS 帯域外の方法により検証するクライアント側に用意する必要がある。安全のため鍵は定期的に入れ替える必要があり、そのため、クライアントが既存の公開鍵の情報を用いてオンラインで新しい公開鍵を取得する方法について規格が定められている。しかし、機器が例えば長期間店頭在庫として保存されていた場合など、長期間オフラインになっていたとき、新しい鍵への交換ができなくなってしまう。このとき、すべての署名されたレコードは検証できなくなるためキャッシュもできず、DNS のトラフィックが爆発することが予想される。本研究では、そのような正当な鍵を所持できなくなった検証サーバの動作を調査し、またその際の解決手法について提案してテストベッドでの検証をおこなった。

Analysis and Treatment of DNSSEC Validator which Own Expired Root Zone Key

Tomohiro Ishihara,[†] Hiroaki Hazeyama[††] and Yuji Sekiya[†]

This is abstract

1. はじめに

Domain Name System(DNS) はドメイン名と IP アドレス等のレコードの対応を示す名前解決システムであり、インターネットのあらゆる通信に先立って使われる。しかし、DNS はその重要性に比べてプロトコルの安全性に問題があり、容易に詐称することが可能である。

そこで DNS のセキュリティ上の問題を解決するために DNSSEC が考案された。DNSSEC は公開鍵暗号を利用してレコードに署名を施すことで、レコードの正当性を保証する。

DNSSEC の公開鍵と署名されたレコードは誰でも自由にアクセスし検証することができるため、DNSSEC の運用においては、攻撃者による鍵の解析を防ぐため定期的に鍵の交換をおこなうことが推奨されている。

安全な鍵の交換のため、DNSSEC では現在の鍵を利用し、新しい鍵の署名をすることで、新しい鍵の正当性を保証する方式[?)] が規定されている。DNS および DNSSEC のレコードを提供する権威サーバでは、新しい鍵の追加に 30 日間、古い鍵の削除に 30 日間の併せて 60 日間を交換時期にあてている。これらの期間について、新しい鍵データを古い鍵で署名した署名データを提供する。期間を経過すると古い鍵を消すことになる。

1.1 DNSSEC トラストアンカー

DNSSEC では、ドメインごとに固有の公開鍵・秘密鍵の鍵対を利用してドメインデータの署名を行うとともに、そのあ下位ドメインの公開鍵を署名することで、下位ドメインの鍵の正当性を保証し、それによって下位ドメインの署名されたデータの正当性を保証する。このように上位のドメインが下位ドメインの保証をおこない、それを連鎖的におこなうことによって、あるドメイン以下のすべてのサブドメインの保証をおこなう。この仕組みを信頼の連鎖（Chain of Trust）と呼ぶ。

クライアントはこの信頼の連鎖をたどることによって DNS レコードの正当性を検証するが、その信頼の連鎖の根本に当たるドメインの公開鍵を前もって取得・保持していなければ検証はできない。この前もって保

† 東京大学
the University of Tokyo
†† 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

ⓒ 2013 Information Processing Society of Japan

持する公開鍵のことをトラストアンカーと呼ぶ。

トラストアンカーは DNSSEC の正当性を保証するものであるため、DNS プロトコル外の方法によって検証を行うクライアントに配布をする必要がある。配布方法としては、例えば物理メディアの受け渡しや公報・書籍への掲載などが考えられるが、より一般的と思われる配布方法としては DNSSEC 検証クライアントが実装されている製品に対してあらかじめトラストアンカーの情報を保存して出荷するという方法が挙げられる。

しかし、前述のとおり、DNSSEC は鍵交換を一定期間でおこない、また一つ前の鍵を利用して鍵の安全な交換をするため、特定の機器が長期間ネットワークに接続されていなかった場合、例えば出荷された製品が在庫として長く店舗で保存されていた場合などでは正当な鍵を取得することができない。このような状態におちいった DNSSEC クライアントは DNS レコードを正当に検証することができず、DNS の問い合わせ結果もキャッシュされなくなる。また、古い鍵の更新の試行を繰り返すため、結果として、多数の問い合わせを出し、権威サーバやネットワークに負荷をかけるようになることが予想される。特に、ルートネームサーバやトップレベルドメインのネームサーバのサーバ及びネットワークにかかる負荷は非常に大きくなる可能性がある。

そこで本研究では、古い鍵のみを保持した DNSSEC クライアントの振る舞いと引き起こす影響をテストベッドを利用した DNS インフラのエミュレーションにより調査し、また発生する問題を解決する手法について提案を行う。

2. 期限切れの鍵を持つ DNSSEC クライアントの動作検証

2.1 検証方針

本現象の検証にあたっては、北陸 StarBED 技術センターのテストベッド環境である StarBED のノードと、米カリフォルニア大 Information Science Institute のテストベッド環境である DeterLab のノードを利用し、各ノード上で DNS サーバ実装が動いている仮想サーバを複数動作させることで仮想的な DNS インフラを構築し、その環境下で失効した鍵を持つ DNSSEC 検証クライアントを導入することで動作確認をおこなった。

2.2 検証環境

図 1 に検証に利用したノードの環境を示す。StarBED のノードを 18 台、DetarLab のノードを 5 台利用した。実験用のセグメントを StarBED 側と DeterLab 側にそれぞれ用意し、VPN により相互接続されたルータを介して双方のセグメントを接続している。また、別途管理用に L2VPN を介して同一セグメントも用意した。

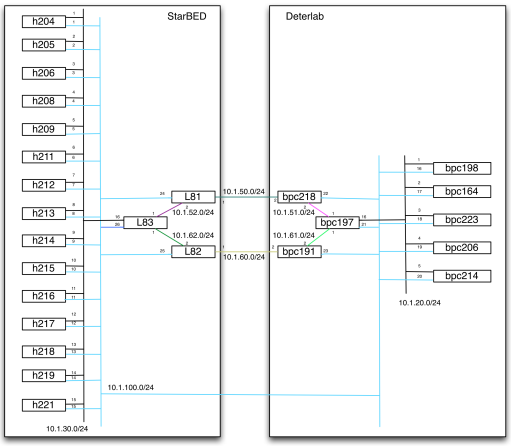

図 1 検証をおこなったテストベッド環境

それぞれのノードは仮想環境のハイパーバイザとして動作させ、仮想環境としては kvm を利用した。

検証に利用した仮想サーバは 50 台の DNSSEC 対応権威 DNS サーバ、34 台の正規のトラストアンカーを持ちネットワークに接続されている DNSSEC 検証クライアント（以下有効検証サーバ）、10 台のネットワークから一定期間切り離される DNSSEC 検証クライアント（以下失効検証サーバ）、および 1 台の鍵更新サーバである。DNSSEC 対応権威サーバとして BIND バージョン 9.8.3 を利用し、DNSSEC 検証クライアントとしては unbound バージョン 1.4.16-1 を利用した。

2.3 実験用 DNS インフラ

実験で問い合わせ対象となる DNS ツリーについては、実験用に独自のルートネームサーバを建て、ルート以下に最大 4 段の深さを持つ 49 ノードのドメインネームツリーを構成し、各ドメインについてそれぞれ 1 台の仮想サーバが権威 DNS サーバを担当した。すべてのドメインは DNSSEC によって署名され、それぞれの上位ドメインの鍵で DNSSEC の鍵を署名することにより信頼の連鎖を構成した。

また、ルートの公開鍵をトラストアンカーとし、それを有効検証サーバ・失効検証サーバに配布した。

2.4 検証手法

まず、すべての有効検証サーバ・失効検証サーバにおいて DNSSEC の検証をおこない、正しく DNSSEC

ⓒ 2013 Information Processing Society of Japan

での正当性検証がおこなえることを確認する。

その後、失効検証サーバをネットワークから切り離し、新しいルート鍵の追加をおこなった。NTP で時間を 30 日進めて、各有効検証サーバで正しく新しいトラストアンカーがインポートされることを確認する。

次に古いルート鍵の削除をおこない、同じように NTP を利用して 30 日進めて、各有効検証サーバで正しく古いトラストアンカーが削除されることを確認する。

トラストアンカーが完全に入れ替わった状態で失効検証サーバをネットワークに接続する。有効検証サーバ・失効検証サーバの双方で実験ドメイン上の名前を DNS で問い合わせるスクリプトを動作させ、その挙動を確認する。

2.5 検 証 結 果

検証環境において、DNSSEC の署名と、有効検証サーバでの鍵更新は正しくおこなえることが確認できた。

表 2.5 は、失効検証サーバが接続された時点での権威サーバに送られたクエリ数の時間あたり量の比較である。

	総クエリ	DNSKEY クエリ
失効検証サーバ	62.86	2.66
有効検証サーバ	3.44	0.13

表 1 権威サーバに送られたクエリ（query/sec）

鍵が失効した検証サーバは、鍵が有効な検証サーバの約 20 倍のクエリを出すことがわかった。また、DNSKEY 問い合わせも同様に増加している。

3. 解決手法の提案

本研究で示した問い合わせの増加は、失効した鍵を持つクライアントが DNSSEC で署名されたゾーンの検証ができないことに起因する。そのため、何らかの方法で有効な鍵をクライアントにインストール必要がある。

ネットワーク機器に新しい鍵ファイルをインストールする方法として、まずファームウェアのアップデートに付随して鍵ファイルをインストールすることが考えられる。しかし、これには以下の問題がある。

(1) 多くのアップデートは HTTP/FTP などのファイル転送プロトコルを利用しており、その際に多くの場合ドメイン名による名前解決をおこなう。正しいトラストアンカーを持っていない状態ではドメイン名が検証・解決できないため、ファームウェアを取得することができない可能性がある。

(2) ファームウェアのアップデートは再起動等を必要とするため、基本的に自動でおこなわない。ユーザが手動でアップデート操作をする必要がある。

そこで、解決手法として、1)DNS の名前解決を利用せず、2) 鍵ファイルのみの取得・検証をおこなう、の 2 点を満たす方法を提案する。

3.1 トラストアンカー更新用鍵による検証

本方式では、各製品ベンダはトラストアンカーを配布するためのサーバを用意する。また、トラストアンカーの検証のためだけに利用する公開鍵暗号方式の鍵対を用意し、各 DNSSEC 対応クライアントには出荷時にその公開鍵を保存させる。また、配布サーバの IP アドレスも同様に出荷時に各製品に設定として埋め込む。サーバはトラストアンカーとして設定したドメインの鍵が更新されるたびに新しい鍵を追加し、トラストアンカー更新用の秘密鍵で署名をおこなう。

クライアントは鍵が更新できないことを検知した時に、埋め込まれた IP アドレスを利用してトラストアンカー配布サーバより新しい鍵とその署名を取得し、保持しているトラストアンカー更新用の公開鍵で署名の検証をおこなう。検証内容が正しければ、新しく取得した鍵をトラストアンカーとして利用する。

3.2 鍵履歴を利用した検証

本方式では、権威サーバ上で過去の鍵をすべて保持し、1 つ世代が新しいものを署名しそのデータもレコードとして公開する（図**??**）。過去の鍵しか所持していないクライアントはその鍵より 1 つずつ新しい鍵を取得・検証することで最新の鍵の入手をすることができる。

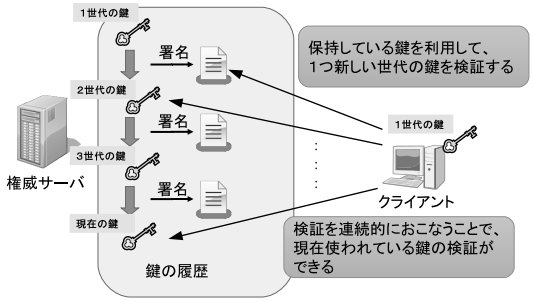

図 2 鍵履歴を利用した検証

3.3 各方式の考察

運用性について、トラストアンカー更新用鍵を利用

ⓒ 2013 Information Processing Society of Japan

した方式の場合、サーバについて IP アドレスを固定の状態で運用し続ける必要があるため、サーバ側におけるネットワーク構成変更に対して柔軟でないという問題点がある。鍵履歴を利用する場合、権威サーバ上でデータを保持する必要があるため、トラストアンカーの場所 (例えばルートネームサーバや gTLD 等) によっては対応が難しいという問題がある。

安全性については、トラストアンカー更新鍵を利用する場合、長期間にわたって同一の鍵を利用する必要があることと、十分に長い鍵を利用する必要がある。頻繁に使われるものでは無いため、長い鍵を利用してもネットワーク・計算資源上のパフォーマンスに影響は出にくいと考えられる。

鍵履歴を利用する場合には、鍵履歴のどこか一点で鍵の流出や解読をされた場合に、その点以前の鍵しか持っていない失効クライアントは不正なトラストアンカーに誘導される危険性がある (図 3)。そのため、だた一つの鍵を対象に攻撃をすればいいため、一つの鍵を利用し続けるトラストアンカー更新鍵と比較した場合、鍵長が同じならば総当り攻撃などで解析される危険性は同じとなる。

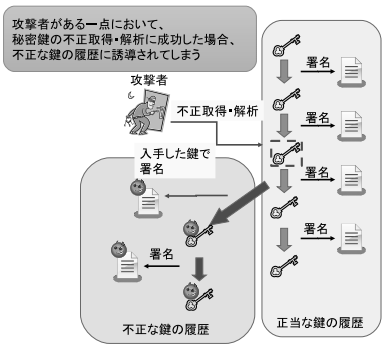

図 3 不正な鍵履歴への誘導

加えて、鍵も通常の DNSSEC で利用される鍵であるため、トラストアンカー更新鍵を利用する場合に比べて長い鍵を使うことが難しい。また、この鍵ではトラストアンカーより高い頻度で定期的に入れ替えられる DNS のゾーン鍵レコードなどを署名するため、署名する対象がトラストアンカー更新鍵よりも多くなり、結果として攻撃者は多くの平文と署名を手に入れることができる。公開鍵暗号方式のアルゴリズムによっては、多数の平文と署名を手に入れることで効率的に攻撃ができる場合があるため、その分のリスクも増加すると考えられる。

総合すると、鍵履歴を利用する方式のほうが、トラストアンカー更新鍵を利用する場合に比べてリスクが高いと考えられる。

4. まとめと今後の課題

本稿では、DNSSEC を利用するクライアントが、長期間オフラインであったことによって鍵がアップデートできずに失効した際に発生するトラフィックの増加について、テストベッド環境において仮想的な DNS インフラを作成することで検証をおこなった。また、検証結果を元に解決手法を提案し、定性的な考察をおこなった。

今後は、提案した解決手法について実証実験をおこない、その妥当性について検証する予定である。

参 考 文 献

1) Peter B. Danzig, Katia Obraczka, Anant Kumar ”An Analysis of Wide-Area Name Server Traffic: A study of the Internet Domain Name System”, *Proceedings of the ACM SIGCOMM*, Baltimore, MD, Aug 1992, pp. 281-292
2) J. Jung, E. Sit, H. Balakrishnan, and R. Morris. ”DNS Performance and the Effectiveness of Caching”, *ACM SIGCOMM* Internet Measurement Workshop 2001,

ⓒ 2013 Information Processing Society of Japan