第６部

特集6 Yeti - A Live Root-DNS System Testbed

加藤 朗

第１章　背景

インターネットは分散的に運用されているネットワークであるが、まったくボトルネックがないわけではない。たとえば、Googleがいきなり使えなくなった場合、多くの人はインターネットをいつもの様に活用することはできなくなる。

DNSは分散的に稼働している名前解決システムであり、きっちりとした同期を諦め、timeoutが発生するまで取得した情報を再利用できるという、緩い結合によって稼働しているシステムである。ここにも、ボトルネックは存在する。DNSの階層的に定義された名前空間の"Root"の解決を担当するRoot DNSサーバがそれである。このノードは、A 〜 M.root-servers.netの13の名前のサーバのいずれかでその部分の名前解決は行われる。現在、Root DNSサーバはAnycast [40, 41] を用いて運用されており、一つの名前で2 〜 199のクラスタがインターネット上で地理的な分散を伴って運用されている*1。

ところで、近年ではIPv6やDNSSECの導入により、DNSにも大きな変化が発生した。これらに限らず、最近の変更は特に応答メッセージが大きくなる傾向にあり、IPv6のdefaultの最小MTUである1280byteを越えることはもちろん、EthernetのMTUである1500byteに近づいている。

インターネットが教科書どおりに稼働しているとすれば、メッセージが大きくなってもフラグメント化によって対応できる。IPv6ではルータによるフラグメント化こそサポートされていないものの、end-to-endでのフラグメント化はきちんと定義されており、問題はないはずである。しかしながら、トンネルの存在やフラグメント化されたパケットの処理がサポートされていない不完全なセキュリティ装置も存在する。つまり、インターネット全域が必ずしも教科書通りに動作するわけではないことが知られている。

Root DNSサーバに対する変更も、Labテストを綿密に実行したとしても、現場に見られるような装置、ソフトウェアのバージョン、動作環境や設定をすべてカバーできるわけではない。もしある変更によってRoot DNSシステムからの応答が届かないようになれば、そのFull-Service Resolver（いわゆるCaching Serverあるいは Recursive Server）はRoot DNSサーバへのアクセスを失うことになる。Cacheがexpireすると、そのFull-Serivce Resolverに依存していたクライアントは、インターネットへのアクセスが事実上できなくなる。

このような状況に対して、live Root DNS testbedを構築し、それによっていろいろな拡張の妥当性をチェックできることが、2015年の春のWIDE合宿中に開催されたunconference sessionで、参加していたPaul Vixie博士およびDavey Song博士、加藤、石原らによって議論された。これを受けて、2015年5月から構築が始まったtestbedは、Yeti Projectと呼ばれている。

第２章　制約

Root DNSサーバのテストベッドとして機能するためには、

＊1　http://www.root-servers.org/

・独自のRootサーバを含むRoot Zoneを生成し、それに基づいて応答すること。
・参加するFull-Service Resolverは、root.cacheファイルをYeti Project版に置き換えること。
・DNSSECのtrust-anchorをYeti Project版に置き換えること。

の３つの条件が必要である。このことは、独自のRoot Zoneを運用する、いわゆる"alternate root" とまったく同じである。

Alternate Rootではないための条件として、

1. インターネット上のRoot Zone（これを特に区別する場合にはIANA Root Zoneと呼ぶ）に対する変更は必要最小限にする。特に、TLDに関するdelegation情報は、IANA Root Zoneのものと同じにする。

2. あくまでも実験であるため、３年間と期限を決めて運用する。つまり、2018年12月末で運用を停止する[*2]。

ということにして、起こりうる懸念を低減させている。

上の条件1. によって、Yeti ProjectのRoot DNSサーバを使っても、通常の名前解決は問題なくできる。Yeti ProjectはProduction Serviceではなくテストベッドであるため、名前解決ができないような事象も発生しうる。そのため、大学や企業のProduction SystemをYetiに依存するようにすることは推奨できないが、研究室などの小規模で、問題が発生した時に自力で対応可能な環境では、Yeti Rootを使うことも可能になる。

第３章　初期構成

Yeti Projectにおける最初の興味は、Root DNSシステムがIPv6だけで問題ないか、Root DNSサーバの名前の数はIANA Rootでは13だが、これより大きくしても大丈夫か、ということであった。そのため、管理上のアクセスはともかく、Yeti RootサーバはIPv6アドレスのみを公開した。また、IANA Rootサーバでは、情報の圧縮を可能にするため、root-servers.netという共通のドメインを使っているが、敢えてそうせず、参加組織のドメインを使うことにした。WIDE Projectで運用しているYeti Rootサーバは、yeti-ns.wide.ad.jpという名前で、そのIPv6アドレスは2001:200:1d9::35である。

Yeti Projectの進行によってYeti Rootサーバの数は変化しているが、もっとも多いときで25あった。現在は22であり、このときのpriming queryの応答は、DNSSECを使用しない（EDNS0のED bitがoff）場合で1318byte、DNSSECを用いる場合には1604byteになる。Yeti RootサーバのOSやソフトウェアにもよるが、1318byteの応答は1280byteを越えるためフラグメント化された応答を返すサーバや、IPv6といえでも1500byteまではフラグメント化しないため、DNSSECを使う場合にフラグメントするサーバが混在している。

第４章　Zone署名鍵と署名処理の分散化

DNSSECはDNSのセキュリティの向上のため、つまりは、cache poisoning攻撃によって不正な情報をcacheに注入されたとしても、署名を検証することによって不正な情報を検出し、誤ったアクセスを防止することを目的としている。このことはよいが、DNSSECの署名およびZoneの分散化は容易ではない。同一組織で秘密鍵を共有した地理的分散は可能であるが、複数の組織で秘密鍵を共有することは現実的ではないためである。これはRoot Zoneのように、信頼性のボトルネックになりうる場合には問題である。

Yeti Projectでは、鍵署名鍵（KSK）の分散化は難しいとしながら、Zone署名鍵（ZSK）の分散化は可能であると考えた。ここで、Zoneの生成および生成されたZoneをYeti Rootサーバに配布する仕事を担当するサーバをDistribution Master（DM）と呼んでいる。DMを複数運用することで、Root Zone生成に関するresiliency向上を図っている。

＊２　この期限に関しては、実験項目のスケジュールの関係で１年間延長し、2019年12月末、とすることが2018年に合意されている。

このDM分散化は以下の様に実装される。まず、署名に用いる鍵は以下のような手順に従って生成、共有する：

・KSKの秘密鍵、公開鍵をDMで秘密裏に共有する。
・各DMではZSKを定期的に生成し（４週間に一度）、相互に交換する。現在は容易のため、ZSKの秘密鍵および公開鍵の両方を秘密裏に共有しているが、実際には公開鍵のみを共有すればよい。

このとき、Yeti Root Zoneは以下の様に生成される。なお、DMはYeti Projectの発起人が準備することにし、BII（中華人民共和国）、TISF（アメリカ合衆国）およびWIDE（日本）がそれぞれ運用している：

・任意のIANA Root DNSサーバからRoot Zoneを入手する。
・. のSOAのMNAME, RNAMEをYeti Projectのものに置き換える。
・全てのDNSSEC関係Resource Record（RR）を取り除く。ただし、DSは残す。つまり、DNSKEYとNSEC、RRSIGが該当する。
・. のNSをYeti Projectのものに置き換える。
・Yeti Projectの. に対するKSKの公開鍵をDNSKEYとして追加する。
・３つのDM運用担当組織で生成した. に対するZSKを、IANAのものと置き換える。
・それぞれのDMで生成したZSK でZoneを署名し、公開する。

なお、上記プロセスは、BII DMが毎時00分、WIDE DMが毎時20分、TISF DMが毎時40分とし、IANA Root ZoneのSOAのserialが変更されていた場合に限り、その以下の手順を実行する。また、その際、RNAMEはZoneを生成・署名したDMが分かるように、SOA のRNAMEにDM名を埋め込むことにしている。

この分散署名方式では、実際にアクセスしたFull-Service Resolverが、どのYeti Root DNSサーバを使用するかによって、RRの署名の検証に必要なDNSKEYが異なる。場合によっては、キャッシュなどの関係で３つのZSKがないと全てのDNSSECの検証ができないことも予想される。しかし、どのRoot Zoneにも３つのZSK全てが含まれているため、DNSSECの検証は失敗しない。Yeti Projectでは、Yeti Rootサーバへの通信の不良などの問題は報告されていたが、DNSSECの署名が不能になるという事象は報告されていない。

第５章　Zone署名鍵と署名処理の分散化（version 2）

分散署名は非常にうまくいったが、全てのRRを再度署名する必要あった。IANA Root Zoneとの、意味的な相違はほとんどないとしても、全てのRRをYeti Projectの対応するDMのZSKで再署名しなければならず、テキスト的には必ずしも差分は小さくなかった。

その後、IANA Root ZoneのTLDへのDelegationに関するRRSIGまで再利用する方法が提案された。この方法では、

・. のNSやKSKに対応したDNSKEY RRは、Yeti Projectのものに置き換える。
・. のZSKに対応したDNSKEY RRはYeti Projectのものに置き換える。ただし、IANA ZSKに対応したDNSKEY RRも残す。
・. に関連するRRを、DMのZSKで署名する。ただし、. のNSECとそのRRSIGはIANAのものを流用する。
・. 以外の全てのRRはIANAのものを流用する。

という様に、. だけがYeti ZSKで署名されており、IANA ZSKも含めてRRSIGはYetiのZSKから生成されている。そのため、Yeti KSKの正当性が検証できれば、IANA ZSKもRRSIGに含まれているため、正当であることが分かる。そのため、一般のJPなどのdelegationに関しては、IANA ZSKによって生成されたRRSIGがそのまま使える、という仕組みである。

第６章　Yeti Projectの今後の予定

Yeti Projectの成果はRFC8483 [42] として2018年10月にIETFのWorking Groupとは独立のindividual RFCとして出版された。また、Yeti Projectでは毎年一回、IETFなど

の際に一日のWorkshopを開催し、関係者とface to faceで議論する機会を得ている。2018年は残念ながらスケジュールの関係で開催できなかったが、2019年3月2日に慶應義塾大学三田キャンパスで開催する予定である。

Workshopのときに議論する予定であるが、まだ完了していない実験項目として、以下のようなものがある：

6.1 Rootサーバ名の署名

IANAのRoot DNSサーバは.root-servers.netという共通のドメイン名を持っているが、このroot-servers.netはDNSSECで署名されていない。これは、署名による副作用、特に世界的に使われているDNSソフトウェアでは問題にはならないような問題の発生を懸念してのことである。Yeti Projectでは、RFC8483には全ての名前はDNSSECで正当性を検証できると書いてあるものの、実際には２つの名前がDNSSECでの署名の対象になっていない。その一つがyeti-ns.wide.ad.jpである。WIDE ProjectのDNS Working Groupによれば、2019年3月合宿を目標にDNSSECでの署名を実施することにしており、近い将来この問題はなくなる予定である。

6.2 Rootサーバ名

現在のYeti ProjectのRootサーバはyeti-ns.wide.ad.jpのような、参加者のドメインをばらばらに使っている。TLDとしては.netが10と最も多く、.inと.comが二つずつ、その他、.at、.ch、.cl、.jp、.nl、.org、.ruが一つずつである。そのため、特定のTLDの名前解決が不能になっても全体の名前解決ができなくなる可能性は大きくない。現在進行中のIANA Rootサーバ名の議論が収束すれば、その妥当性に関するチェックをする可能性もある。

6.3 アルゴリズムRoll Over

DNSSECで面倒なのは、鍵の交換である。ZSKは半自動で生成や置き換えが可能であり、IANA Rootでも３ヵ月に１回程度はZSKの更新を実施している。しかし、Root Zoneに関して言えば、KSKはtrust anchorとしての意味があり、多くのDNSSEC対応のfull-Service Resolverでは、それを設定する必要がある。RFC5011をサポートしているソフトウェアで長時間インターネットから切り離されることがなければ、新しいKSKの取得と古いKSKからの承認を自動的に行うことができる。現在IANA Rootでは最初のKSK Roll Overが実施されており、本文執筆時点ではその最終段階の一つ手前、つまり、2010年から使われている鍵KSK-2010にrevocation bitを立てた段階で、2019年3月頃、KSK-2010はRoot Zoneから姿を消すことになる。

一方、アルゴリズムRoll Overは、鍵や署名に使われているアルゴリズムを変更することである。現在のIANA Rootは、鍵や署名に関してはRSASHA256を用いている。もしRSASHA256が危殆化した場合には、より頑強なアルゴリズムへの変更が必要である。RFC4034 [43] で必須として定義されているのはSHA-1だけであるが、RFC6944 [44] では、RSASHA256、RSASHA1-NSEC3-SHA1、RSASHA512、ECDSAP256SHA256、およびECDSAP384SHA384が\Recommended to Implement" となっている。さらにRFC8080 [45] ではED25519およびED448も提案されている。

事実上全てのDNSSEC実装でサポートされているアルゴリズム間の移行は、新旧両方のアルゴリズムを用いた鍵や署名をサポートしなければならないため、DNSの問い合わせパケットが肥大する。また、全てのDNSSEC実装でサポートされているとは言えないアルゴリズムへの移行は、この併用期間を数年程度以上に設定する必要があり、パケット長問題が深刻になる可能性がある。

そのため、Yeti ProjectでアルゴリズムRoll Overを実施するかどうかの結論はまだ得られてはいないが、起こりうる副作用やその確認方法も含めての検討が必要である。2019年末までのYeti Projectの期間に終了しない可能性もあり、この場合には再度の延長も考慮しなければならない可能性もあることを記しておく。