DNSSECのシステム上の実現課題とトランスポートに関する妥当性検証

力武 健次
情報通信研究機構 インシデント対策グループ
2006年12月6日
Internet Week 2006 dnsops-jp BoF

DNSSECの必要性と展開状況

- DNSSEC: RRの認証→ドメイン名詐称防止
- 必要なDNSの拡張
  - RFC4033/4034/4035: DS, RRSIGなどの追加
  - トランスポート層の拡張 (EDNS0)
  - Key rollover等認証の一貫性維持
  - リゾルバ，サーバ，キャッシュすべて対応が必要
- ルートサーバでの実装を目指して作業中
  - 一部TLD (.seなど) で実装実験中

DNSシステム仕様上の課題

- CPU資源，所要主記憶量
- 署名による処理量の増大
  - ゾーン署名，鍵用の乱数生成などの作業
- トランスポートで発生する問題
  - 必要通信帯域とペイロード長の増加
  - 512バイトを超えた場合のTCP/UDPの選択
  - UDPペイロード長増加→IP fragmentation
  - fragmentation許可にend nodeが対応していない

CPU資源とメモリ消費量

- CPU資源の増加量: 3倍以下
  - 署名の検証，追加ペイロード生成，パケット処理
  - authoritative serverで2倍，キャッシュで3倍程度
  - サーバやキャッシュの増設で対応可能
- メモリ消費量の増加: 2〜5倍程度
  - 追加ペイロードが消費するメモリ量が主に関与
  - authoritative serverの例: 156MB→290MB
  - キャッシュの例: 93MB→432MB

ゾーン署名に必要な計算量

- 各ゾーンはRRSIG RRの追加作業が必要
- 署名処理時間は鍵長の4乗にほぼ比例
- .caでの例 (2005年12月，612kレコード)
  - P3/1.4GHz 3GBメモリ Linux 2.6 Kernelでテスト
  - 63MB→238MB (1024bit)，300MB (1584bit)
  - 署名に29分 (1024bit)，85分 (1584bit)
  - ゾーン情報5分割で29分→9.5分まで短縮

鍵生成に必要な乱数生成量

- Key Signing Key (KSK)
  - ZSKを含むDNSKEY RRのみを署名
  - 1024〜2048bit，13ヶ月で更新
- Zone Signing Key
  - KSKとの長さの差は100bit未満であること
  - 1ヶ月で更新
- .comでも市販の乱数生成装置で対応できる
  - 毎秒88ゾーン→88kB/sec << 10〜100Mbps

通信量の増加による トランスポート問題

- DNSSECはUDP上の大きなペイロードを使う
  - 従来は512バイトが最大
  - EDNS0拡張で4k～8kB/ペイロードとなる
- 帯域は2～5倍に増大との予想あり
  - DNSSECの比率50%で3.6倍，90%で4.8倍 (NIST)
  - authoritative serverでは2～3倍 (Kolkman)
- ペイロード長増大→IP fragmentation
  - IPv4 MTU: 1500bytes (ethernet)
  - 到達性の低下など伝統的な課題が**広域で発生**

ペイロード長増加とその影響

- IPパケット1個では運べない→fragmentation
- fragmentationの発生確率予測
  - IPv6では30%，IPv4では15%（力武他）
  - IPv4では77% （Ager他，NXDomainを含む）
    - NXDomain: 2xNSEC, DNSKEYと署名が追加
- 2つの対策方法
  - DNSKEY+RRSIG を送らない（nsdで採用）
  - Elliptic Curve Keysなど高効率なハッシュを採用
    - 現実にはRSA/SHA1やRSA/SHA-256が標準

TCPへのフォールバックと IP fragmentationの問題

- TCPへのフォールバックによる問題
  - UDP最大許容ペイロード長の設定間違いで発生
    - 従前の512バイトの上限値をそのまま採用
  - ゾーンが署名された途端にTCPのqueryが発生
    - BIND 8ではUDPで取れないとすぐにTCPで再試行
- IP fragmentationを扱えないホストとルータ
  - **実運用ではfragmentパケット通過を禁止**している
    - **これでも従来のDNSやTCPアプリケーションは動く**
  - 問題はバックボーンではなく**leaf nodesの対応**
    - **末端で通過拒否の場合DNSSECは利用不能**

トランスポートの妥当性検証(1): 大きなRRに対するqueryの実験

- 大きなTXT RRをサーバに置いてもらう
  - 2550byteのRDATAフィールドを持つものを用意
  - 単一のRRは分割されない
- このRRに対するqueryを送信
  - 受信できれば運用上は問題なし
- 問題点: 事前にTXT RRを用意する必要あり
  - DoS攻撃に使われる可能性がないとはいえない

トランスポートの妥当性検証(2): 大きなUDP queryの送信

- 大きなUDP queryを送信側で送ってみる
  - サーバから反応があれば通っていると考えてよい
  - IP fragmentsが揃わないとUDPは通らない
- 実際には単独のquery+ダミーで可
  - 最初に有効なqueryとEDNS0のOPT RRを置く
  - その後はダミーで埋めて2048バイトにする
  - サーバはqueryの有効部分しか見ない（BIND）
- 問題点: 仕様外の使い方→動作保証なし
  - サーバが異常検知して落とされるかも

結論と今後の課題

- DNSSECはシステム資源上は実現可能
- しかしトランスポート上は問題が残っている
  - IP fragmentationの問題は古くからあるが未解決
  - Leaf nodesがfragmentation禁止なのが問題
- Fragmented IP透過の検証手法の確立
  - UDPの特性を使えば検証可能だが問題も残る
- 大規模fragmentation発生時の実験が必要
  - 現状では対応したシミュレータの入手は難しい