

DNS Cache Poisoningの概要と対処

(Dan KaminskyによるDNS脆弱性指摘に関して)

23rd Jul 2008

Updated 19th Aug 2008

NTT情報流通プラットフォーム研究所

豊野 剛

toyono@nttv6.net

はじめに

- この文書の目的
 - 2008年7月22日に漏洩, 8月7日に公開されたDNS脆弱性攻撃手法に関して概略的に述べたものです
 - 問題点と影響度
 - 対処方法
 - ~~– 攻撃手法に関しては説明していません~~
 - BlackHat2008の資料が公開されたため附記 (8/7)
 - 多くのWebサイト上で技術的な解説がされています
 - 各種の攻撃ツールが出回っています

経緯

- 7月8日 (9日 JST)
 - 各CERTから DNSの脆弱性 (Cache Poisoning攻撃)に関する Alertが発行される
 - 同時に8月に開催されるセキュリティ系会議 (Black Hat 2008 USA)で脆弱性の発見者 Dan Kaminskyが詳細手法を公開する旨を通達 (事実上の対処期限日となるはずだった)
- 7月21日 (22日 JST)
 - Dan Kaminskyの DNS脆弱性攻撃手法を解説した記事が blogで公開される
 - blog公開者は誤った情報公開だったとして同日中に記事を削除する
 - しかし同記事の Webキャッシュ, コピー記事等が既に WWW中に蔓延し, **事実上の脆弱性攻撃手法の公開日**となる
 - 同日, Dan Kaminskyも「脆弱性情報が公開された」とコメントした
- 8月6日 (7日 JST)
 - Black Hat 2008 USAにおいて発見者 Dan Kaminskyが講演

DNSとは

- IPアドレスとドメイン名(FQDN)を相互変換するマッピングデータベース
 - ドメイン名とIPアドレスのいわば電話帳
 - www.ntt.co.jp ←相互変換→ IPアドレス 10.0.0.1
 - 一つのデータベースをインターネット上で分散運用
- ドメイン名で通信する限り、ユーザのアクセスの際に必ず利用される
 - ユーザのアクセサビリティに直結する重要インフラストラクチャ
 - WWW以外にも、メール(MX), ENUMなど、DNSに依存しているサービスは多い

どんな脅威があるのか

- DNSキャッシュサーバにおいて、ドメイン名が乗っ取られる可能性がある
 - 嘘の応答を正規の応答より先に返すことでDNSキャッシュサーバに偽情報をキャッシュさせる (DNS Cache Poisoning攻撃)
 - ドメイン名とIPアドレスの対応付けを横取りされてしまう
 - 正規応答を横取りするため、ユーザは正しい情報との判別ができない
- 本来の宛先ドメイン名に対するあらゆる通信を横取り出来てしまう
 - ドメイン名を用いているあらゆるサービスに対する Phishingが可能
 - サービス不能攻撃
 - メールなどのトラフィック横取り
 - 個人情報, アカウント, パスワードなどの略取
 - 著名サービスのドメイン名が乗っ取られると極めて危険度が高い
 - 証券, 金融, 銀行系などの Phishingでは直接金銭被害に結びつく可能性もある

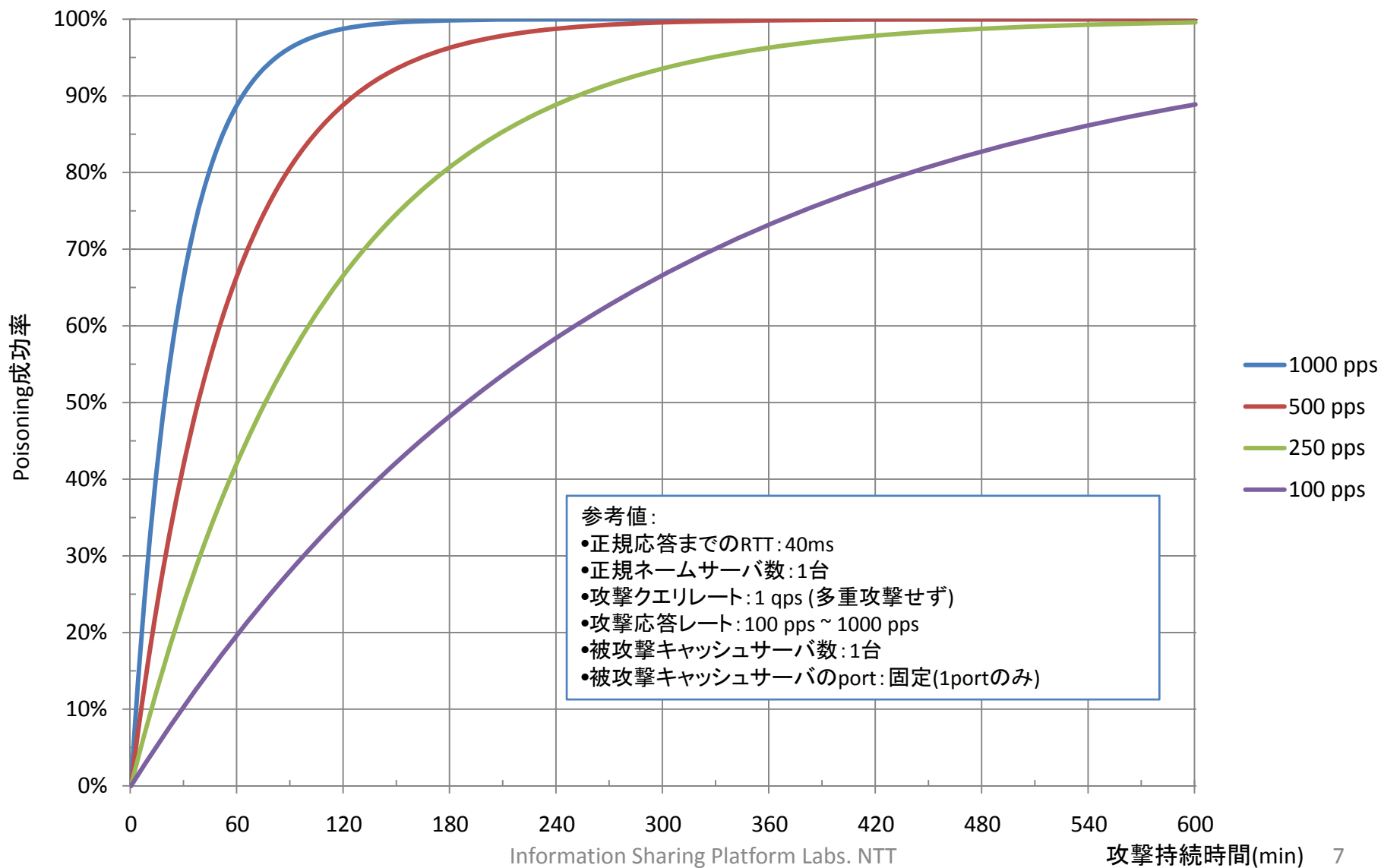
攻撃の成功率はどのくらいなのか (1/3)

- Cache Poisoning攻撃は以前から知られていたが、今回の攻撃ではリソースレコード(RR)の Cache TTLが 0秒になったのと同じ攻撃が可能になった
 - 今回の攻撃では正規RRの TTLを無視して連続した攻撃が可能
- (例) 仮に1000pps(秒間1000回の偽応答)の攻撃を許した場合
 - 約19分(1136 秒)で攻撃成功率が50%を超える
 - 約5時間半(271分)で攻撃は必ず成立(100%)
 - 数時間攻撃され続けるとそのドメインは必ず乗っ取られてしまう
 - より高い攻撃レートや、マルチスレッドでの攻撃を受ければ数十秒程度で Poisoningされてしまうことも

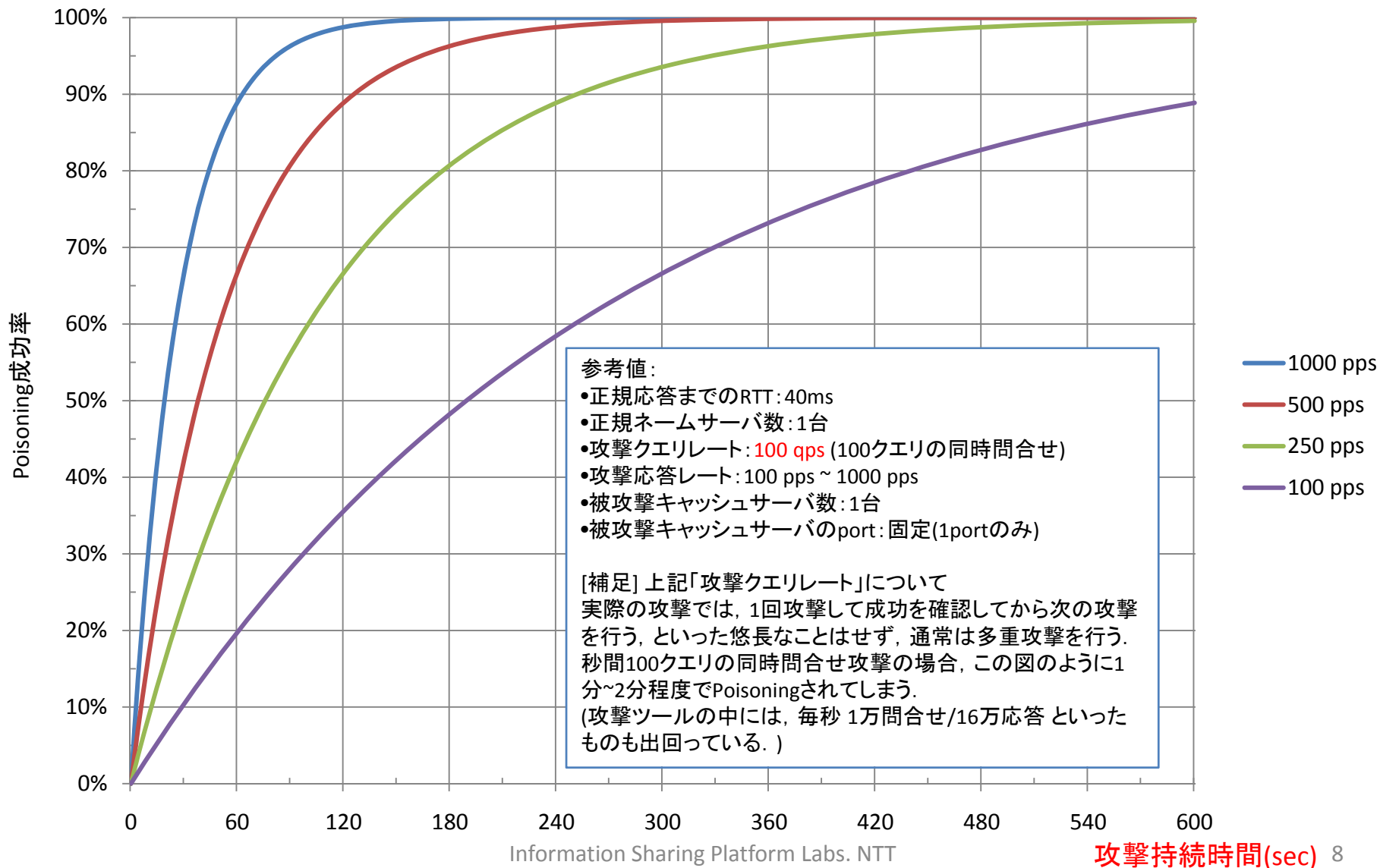
参考値:

- 正規応答までのRTT: 40ms
- 正規ネームサーバ数: 1台
- 攻撃クエリレート: 1 qps (多重攻撃せず)
- 攻撃応答レート: 100 pps ~ 1000 pps
- 被攻撃キャッシュサーバ数: 1台
- 被攻撃キャッシュサーバのport: 固定(1portのみ)

攻撃の成功率はどのくらいなのか (2/3)



攻撃の成功率はどのくらいなのか (3/3)



何が問題を複雑にしているか

- 攻撃に用いられている手法はDNSプロトコル自体のメカニズムに基づいており、**根本的に対処する方法が現状のところ存在しない**
 - 根本的な解決にはデジタル署名を組み込んだ DNSSECを利用するしかないと言われているが、まだ実装・普及が不十分なのが現状
- 攻撃成功確率を低減する対処を講じる必要はある
 - 今回の脆弱性に対する各DNSベンダのパッチ対応
 - DNSの問合せportをランダム利用することで、攻撃が当りづらくしている
 - $\text{DNS TXID}(16\text{bit}) \times \text{port番号}(16\text{bit}) = 32\text{bit}$ 分のランダム性を確保

攻撃確率を軽減する対処方法 (DNSコンテンツサーバ運用者)

- (自分の管理するドメインに対する)ネームサーバの台数を増やす
 - 正しい応答をする可能性のある(正規)ネームサーバを複数にすると、攻撃者は複数ネームサーバ分の応答を偽造しなければならない
 - 成功率が下がるのみで安全な対策とは言えないが、やらないよりは...

攻撃確率を軽減する対処方法 (DNSキャッシュサーバ運用者)

- 今回の脆弱性の対応パッチ(port randomize)を必ず当てる
 - Portを固定する設定が入ってしまうとPatchの意味が無いので注意
 - Configでportの固定設定をしていないかどうか注意する
 - NAT配下の場合など, 別の機器でportが収束してしまわないように注意する
 - 場合によってはDNSキャッシュサーバをNAT配下からDMZへ收容替える必要も有り
- Open Recursive(外部へのCache応答)を止める
 - 攻撃を受けづらくする/内部ネットワークからの攻撃のみに限定する
 - ISP網など管理しきれない第三者が利用している, 内部ネットワーク利用者がBotに感染してしまう, など, 閉じたからといって攻撃が発生しなくなるわけではないことには注意する
 - 外部からの攻撃を受けづらくする意味では, キャッシュサーバとコンテンツサーバを分離して運用することも有効
- 単位時間当たりの単一ネームサーバからの応答クエリ数を制限する
 - 通常は1つの問い合わせに対し, 1つの応答が帰ってくるはず
 - Poisoningを狙われたドメインに対しては大量の偽応答が混じる
 - ただし正常な利用でも単一DNSサーバと多くのトランザクションが発生する場合もあり, クエリ数制限を行う場合には注意の必要有り
 - gTLD, TTLの短いドメイン, サブドメイン数の多いドメインなど

まとめ

- 今回のDNSの脆弱性は...
 - 非常に危険で影響力が大きい
 - きちんとしかるべき対応を取る
 - 今回の対応策は暫定対処であることを理解しておく
- 今後の動向として... (今回のDNS対応に限らずセキュリティ全般に言えることですが...)
 - 新しい攻撃手法が継続的に生み出されています
 - 新しい対策を継続的に講じる必要があります
 - DNSSEC, その他大規模なプロトコル拡張が普及してくるかもしれないので注視しておく

(参考) 参考URI

- 日本語
 - JPCERT/CC
 - <http://www.jpccert.or.jp/at/2008/at080014.txt>
 - JPRS
 - <http://jprs.jp/tech/security/multiple-dns-vuln-cache-poisoning.html>
 - <http://jprs.jp/tech/security/multiple-dns-vuln-cache-poisoning-update.html>
 - JPNIC
 - <http://www.nic.ad.jp/ja/topics/2008/20080709-02.html>
 - JANOG19 “これでいいのかTTL”
 - <http://www.janog.gr.jp/meeting/janog19/2006/12/dnsttl.html>
- 英文
 - US-CERT
 - <http://www.kb.cert.org/vuls/id/800113>
 - Dan Kaminsky氏
 - <http://www.doxpara.com/>
 - 同氏の Black Hat 2008 講演資料
 - <http://www.doxpara.com/?p=1204>
- 漏洩関係(英文)
 - <http://www.matasano.com/log/1105/regarding-the-post-on-chargen-earlier-today/>
 - <http://blog.invisibledenizen.org/2008/07/kaminskys-dns-issue-accidentally-leaked.html>

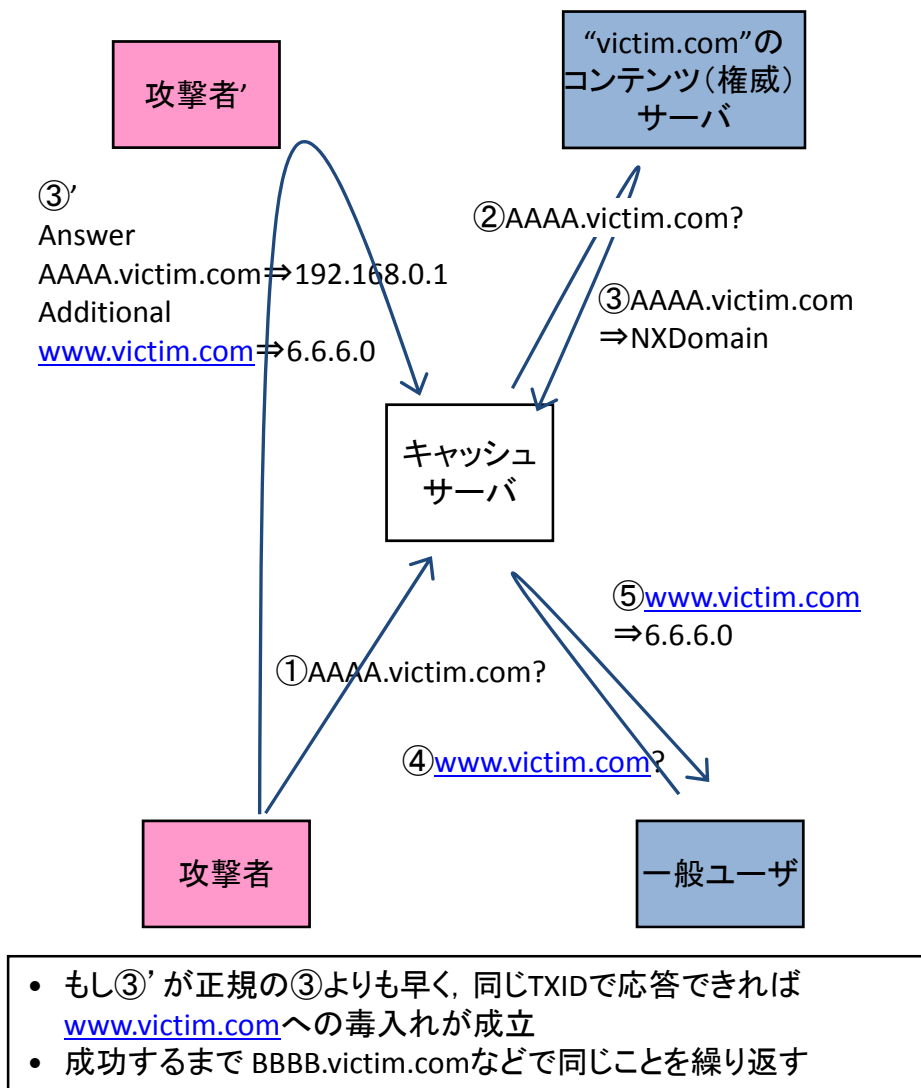
(参考) Dan Kaminsky's Attackの手法概説

• 手法

1. 攻撃者が汚染したいドメイン名と同じゾーンの存在しない適当なドメイン名に対するクエリをキャッシュサーバに送信する (図①)
2. 攻撃者自身が(TXIDを変えながら)レスポンスを多数応答する (Src IPは Spoofing) (図③')
 - 存在しないドメイン名に対するレコードと共に汚染したいドメイン名のレコードを追加セクションに記述
3. 正規権威サーバより先にクエリと同じTXIDのレスポンスをキャッシュサーバに応答できれば汚染が成立する
 - DNSのTXIDは16bitしかない (1/65535で攻撃成功)

• 特徴

- 存在しないドメイン名はいくらでも作れるので攻撃回数を増やせる
 - 全てのレコードがTTL=0秒になったのと同様の連続攻撃が可能
- 攻撃するドメイン名を直接でなくても, NSを追加応答された場合, Zone(NS)ごと汚染されてしまう



(参考) 攻撃が成功する確率

$$P_{(t)} = 1 - \left(1 - P_{(s)}\right)^{t \times Rq}$$
$$= 1 - \left(1 - \frac{Rr \times W}{N \times Port \times ID}\right)^{t \times Rq}$$

P(t): 攻撃成功確率

P(s): 1回のクエリで攻撃が成功する確率

t: 攻撃持続時間

Rr: (1クエリ当たりの)応答攻撃レート

Rq: クエリ攻撃レート

W: 正規応答が帰ってくるまでのRTT

N: 攻撃対象のレコードを保持するネームサーバ数

Port: Query portの数(固定portの場合1)

ID: DNSのID (16bit = 65536)

JANOG19 “これでいいのかTTL” 民田さん@JPRS の公開資料を改変

(参考) 攻撃の成功率はどのくらいなのか Port Randomize後

