変なdig

このオプション、何かがおかしい。

(+trace オプションの謎)

アカマイ・テクノロジーズ合同会社
シニア・ソリューション・エンジニア

松本 陽一

このプレゼンテーションにおいてなされる記述は作成者個人の見解を示すものであり、アカマイ・テクノロジーズの見解を示すものではありません。提供される情報は作成時点において正確なものであると考えておりますが、当該情報についてなんら表明又は保証を行いません。

DNS Summer Day – dig シリーズ第 3 弾

2023 「あたらしい dig」
2024 「あたらしかった dig」(「わたしたちの DNSViz」)
2025 「変な dig」

DNS の調査ツールについて楽しく学んで
- 日々の仕事(調査)に役立てよう
- DNS の知識を広げよう

dig - 新しいオプションを調べて新しい標準を知る

例)

+[no]zoneversion (9.21.7 で追加)

RFC9660 The DNS Zone Version (ZONEVERSION) Option (2024 年 10 月)

クライアントがクエリにこの EDNS オプションを付けると、サーバーはゾーンのバージョン (SOA のシリアル値等) を付けて応答する

2024 年 9 月 – 知人からの相談

第一子が生まれたばかりの彼は一軒家を買う決心をし、「DNS に詳しいから」という理由で、私に相談することに決めたようだ。

「dig +trace で、サーバーを指定しないと動かない場合がある。@1.1.1.1 だと動くけど、@8.8.8.8 だと動かない。」

dig(1)
man ページより

+trace, +notrace

This option toggles tracing of the delegation path from the root name servers for the name being looked up. Tracing is disabled by default. When tracing is enabled, **dig** makes iterative queries to resolve the name being looked up. It follows referrals from the root servers, showing the answer from each server that was used to resolve the lookup.

If **@server** is also specified, it affects only the initial query for the root zone name servers.

+dnssec is set when *+trace* is set, to better emulate the default queries from a name server.

Note that the **delv +ns** option can also be used for tracing the resolution of a name from the root (see *delv*).

+trace

反復的なクエリを実行。ルートサーバーからのリファーラルを追跡し、各サーバーからの応答を表示する

- @server を指定した場合、これはルートゾーンのネームサーバーを問い合わせる最初のクエリにのみ影響する
- +dnssec がセットされ、ネームサーバーからのクエリをより正確にエミュレートする
- delv +ns オプションもルートから名前解決するトレースに使用できる

～ JANOG53 Meeting In Hakata より ～

DNS チュートリアル
(https://www.janog.gr.jp/meeting/janog53/dnsbasic/)

JPRS 森下 泰宏 氏
「つい一番下に書いてある dig +trace という便利なオプションを使いたくなるんですけど、これは実はフルリゾルバー経由で権威 DNS サーバーへの問い合わせをトレースするような形になってますので、とても簡単に言うと靴の上から足を掻いているようなもので、一個だけ権威 DNS サーバー動作おかしいなんていうときにうまく調べられないことがあります。
わたし、実は dig +trace ってもうずっと使ってなかったんです。
あんまり調査にならないんで」

実際の dig +trace の動作を見てみよう
(パケットキャプチャからの推定)

これは何？

1. ルート (.) の NS を @server で指定したサーバーに問い合わせ、結果を表示する
2. 得られた NS レコードの名前全てに関し、IPv4/IPv6 アドレスを解決する (表示はしない)

反復的っぽい

3. そのうちからランダムに選択した一つのアドレスに向けてコマンドラインで指定した QNAME / QTYPE をクエリし結果を表示する (RD=0)
4. リファーラルを受け取った場合、その NS レコードの名前全てに関し、IPv4/IPv6 アドレスを解決し(表示はしない)、3. へ
5. リファーラルではない権威ある応答を得たら表示しておしまい
   CNAME は辿らない

1. ルート (.) の NS を問い合わせる (@server で指定されていればそのサーバー)

プライミングクエリと考えられる

RFC 9609 Initializing a DNS Resolver with Priming Queries

フルサービスリゾルバーに付属する hints の内容が最新とは限らないので、最新のルートサーバーのリストを取得するためのクエリ

プライミングクエリの宛先は、通常はリスト (hints) に載っているルートの IP アドレスから選ばれる

+trace オプションを指定した場合でも hints からは選んでくれず、/etc/resolv.conf やコマンドラインの @ で指定したサーバーに向けられる

フルサービスリゾルバーがプライミングクエリを受けた場合どうなるか

dig +trace の出す QNAME=. / QTYPE=NS / RD=0 のクエリに対し、1.1.1.1 はキャッシュから応答するようだが、8.8.8.8 は SERVFAIL で応答する
→　dig +trace @8.8.8.8 が動かなかった理由

実は 9.11.8, 9.14.3, 9.15.1 の dig では RD=0 であることが原因のバグとして RD=1 になるよう「修正」されている
(https://gitlab.isc.org/isc-projects/bind9/-/merge_requests/1939)
→　@8.8.8.8 でも動くようになっている

ところが RFC9609 (February 2025, Obsoletes RFC8109) では「The Recursion Desired (RD) bit SHOULD be set to 0. 」つまり RD=0 が正しいことが明記された

なお +trace の後に +nordflag (+norecurse) をつけると RD=0 になる

2. / 4. 得られた NS レコードの名前全てに関し、IPv4/IPv6 アドレスを解決する(表示はしない)

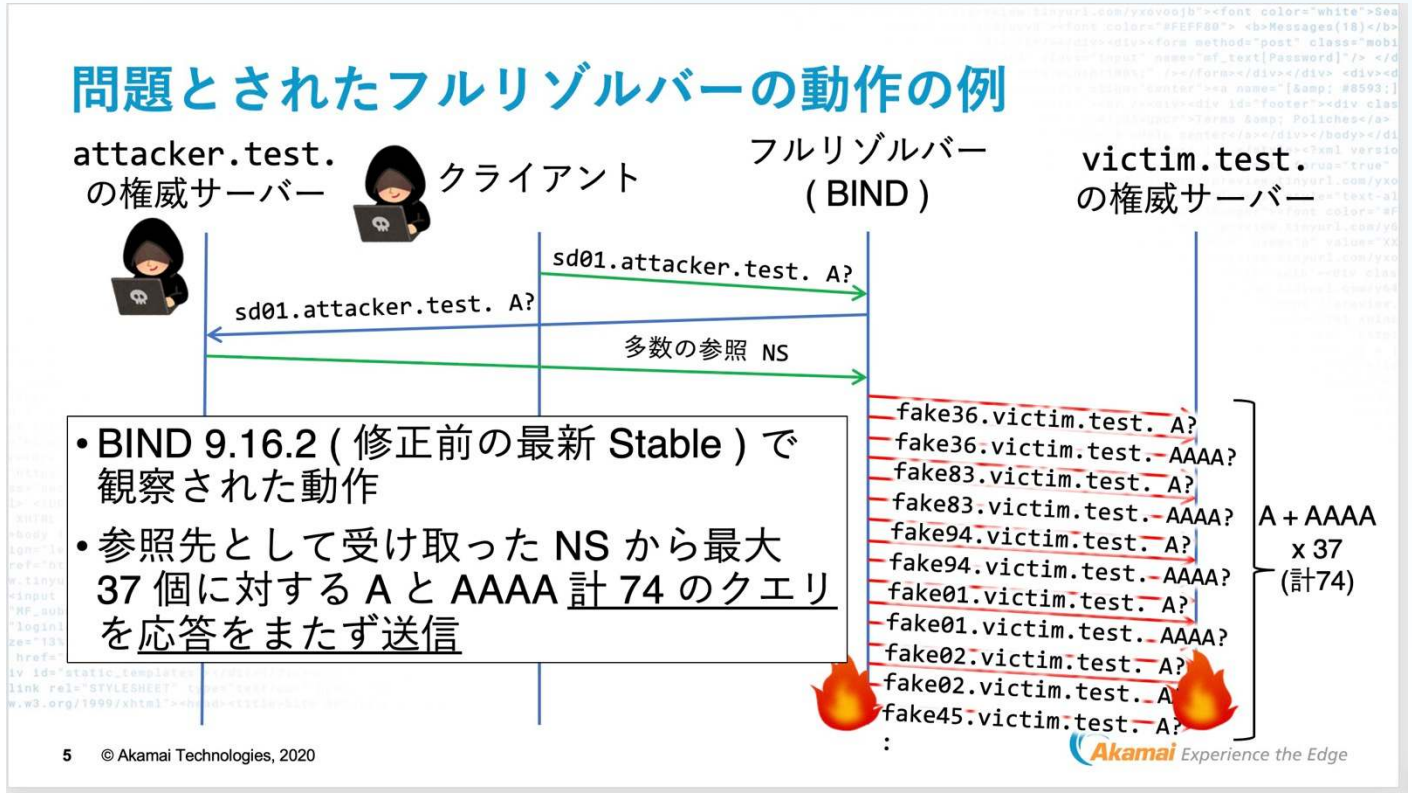
表示はされていないが、例えば 13 個ある .com の NS に A/AAAA、つまり 26 個のクエリを行っている

本来の反復クエリとして解決しているわけではなく、スタブリゾルバー(getaddrinfo())で解決しているだけ
グルーも見ていない 「靴の上から足を掻いているようなもの」

NS で指定された名前の A/AAAA を全部解決すると言えば、確かに NXNSAttack を巡って調べた BIND 9 の動作の通りだが、一つずつ行なっている点も異なる

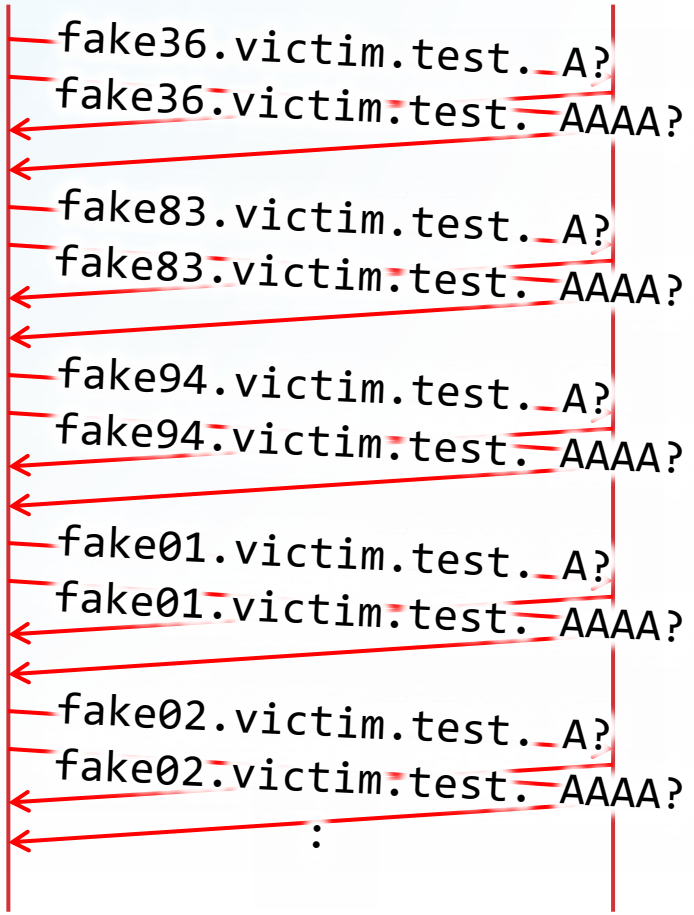
named の反復クエリと dig +trace における NS のアドレス解決の比較

- 宛先が権威ネームサーバーでない
- 一つ一つ解決



dig +trace

OSに設定された
参照フルリゾルバー



オプション指定の上書きは+dnssec だけではない

- +rdflag (+recurse) – 最初のクエリに有効。前述の「修正」による
- +dnssec – BIND 9 では反復クエリの DO フラグは設定に関わらずオン
- +cookie – BIND 9 の反復クエリではデフォルトでオン
- +answer
- +auth
- +identity – どのサーバー/ポート番号からの応答か表示 (実は +short オプションだけでなく、+trace オプションでも有効であったことが判明)
- +nocomments – Answer Section / Authority Section の区別がつかない
  8.8.8.8 から SERVFAIL が返ってきていたことに気づけなかったのもこのせい
- +norrcomments
- +nostat
- +noquestion – 何をクエリした結果なのか表示されない
- +noadd – グルーが表示されない

これらは +trace より後ろにオプションを指定することでさらに上書きできる
+qr や +yaml も有効なので、自分なりに見やすくアレンジしてみよう

まとめ – dig の +trace オプション

- 反復クエリをしているというよりも反復クエリのようなデモをしていると言えそう
- プライミングクエリはルートサーバーに向けられるべきだし RD=0 であるべきなのに…
- リファーラルの NS の名前解決をスタブリゾルバー(getaddrinfo())に頼っており、グルーも見ていない
- デモにしては重要なものを表示せず不親切にも思えるが、+trace を指定することで上書きされるオプションを知ることでいろいろ分かる
- トラブルシューティングで役に立つのかは確かに疑問だが、動作を理解しておいて損はないはず

delv

delv – dig、nslookup、host と共に BIND 9 に付属するツール
Domain Entity Lookup & Validation の略とされる。cf.「delve」 掘り下げる

dig と異なり
- DNSSEC バリデーションを行う
- 指定したサーバーに向けて必要な一連のクエリを行なう
- フルサービスリゾルバーに向けることを想定

つまり、DNSSEC に特化したツールであった

delv +ns

- 「name server mode」に切り替える。delv 自身がフルサービスリゾルバーとして動作するモード
- 9.19.12 / 9.20.0 [GL #3842] で導入
  https://gitlab.isc.org/isc-projects/bind9/-/issues/3842
- +qmin[=MODE] オプションで QNAME Minimisation の切り替えもできる。(ちなみに dig +trace は QNAME Minimisation しない)
- -i オプションで DNSSEC Validation しないようにできる
- DNSSEC の調査にも役立つが、DNSSEC 検証失敗以外の失敗理由も分かる場合がある。
  named のログよりも多くの情報を得られる

比較

	dig +trace	delv +ns
プライミング・クエリの宛先	通常と同じ @による指定または /etc/resolv.conf 等	hints から選択される
プライミング・クエリの RD フラグ	RD=0 だったものが RD=1 に「修正」された	RD=0 (RFC9609に沿う)
リクエストの表示	表示されない (Question Section も表示されない)	表示される
各セクションの表示	セクションの区別がつかず、 Question Section や Additional Section は表示されない	全セクション明記して表示
委任先 NS のアドレス解決	OS の参照先のフルサービスリゾルバーに依存	フルサービスリゾルバーとして反復クエリにより解決
CNAME	辿らない	CNAME を辿って解決
出力の長さ	長い	すごく長い
YAML 出力 (+yaml オプション)	対応	不完全

[GL #3842] delv +ns (name server mode) to replace "dig +trace"

The results would be similar to dig +trace - which has always been rather buggy - but since it would be using the same code as named, it would exactly reproduce the behavior of a name server with a cold cache performing a recursive lookup. dig +trace can be deprecated once this feature is in place.

結果は (従来からバグが多い) dig +trace と似ているだろう。ただし、 named と同じコードを用いるため、空のキャッシュで名前解決を行うネームサーバーの動作を正確に再現するだろう。この機能ができれば dig +trace は非推奨になりうる。

さようなら、dig +trace

配給