

事後公開資料
(発表時の内容から追加・更新あり)

教科書には載っていないDNS

副題: キャッシュDNSサーバーの憂鬱

2013年7月19日

DNS Summer Days 2013

株式会社日本レジストリサービス (JPRS)

森下 泰宏

@OrangeMorishita

講師自己紹介

- 氏名: 森下 泰宏 (もりした やすひろ)
 - 1965年9月21日生まれ (47歳) 男性
 - 勤務先: 株式会社日本レジストリサービス
 - 肩書: 技術広報担当
- 最近の願いごと: 平穏無事な7月
 - 毎年願っては裏切られています



本日の内容

- 昨年、ここでこんな話をしました
 - その中で...



JPRS
JAPAN REGISTRY SERVICES

DNS入門

2012年8月31日
DNS Summer Days 2012
株式会社日本レジストリサービス(JPRS)
森下 泰宏

Copyright © 2012 株式会社日本レジストリサービス 1

DNS Summer Days 2012

「DNS入門」より再掲

重要な注意(1)

- このパートの「②:階層構造をたどる」で説明する「たどるサーバー(=キャッシュDNSサーバー)」の動作は、現時点における**実際のものとは一部異なっています**。
- 特に、外部名／内部名のチェック、RFC 2181で定義されているリソースレコードの信頼度のチェックなど、本来必要となるいくつかの事項について、**意図的に説明を省略**しています。

重要な注意(2)

- その結果、RFC 1034/1035が意図したと考えられるよりシンプルな(ただし、セキュリティ上の考慮が十分ではない)動作説明となっています。
- 今回、意図的に省略した部分の動作説明については、いつか何らかの形(続編(?))で実施できたらいいなと考えています。

本日の内容

- ということで、昨年省略した「名前解決におけるキャッシュDNSサーバーの動作」について、

1. グルーと内部名・外部名
2. 委任情報とreferral
3. グルーはグルー（DNSデータのランキング）
4. カミンスキー型攻撃手法

といったあたりのお話をします

事前の注意

- 本日のお話は、キャッシュDNSサーバーの動作の「**ごく一部**」に過ぎません
- 「キャッシュDNSサーバー山」は**高くて険しい**です
 - 私にもその頂は**見えません**(ようやく一合目ぐらい?)
- 本日のお話は、チュートリアルとしては少し難しいかもしれません(ごめんなさい)
 - 「digコマンドの出力を読める」を前提にしています
- 本日のお話で、「**キャッシュDNSサーバーの憂鬱**」を、みなさまに少しでも共感いただければ幸いです

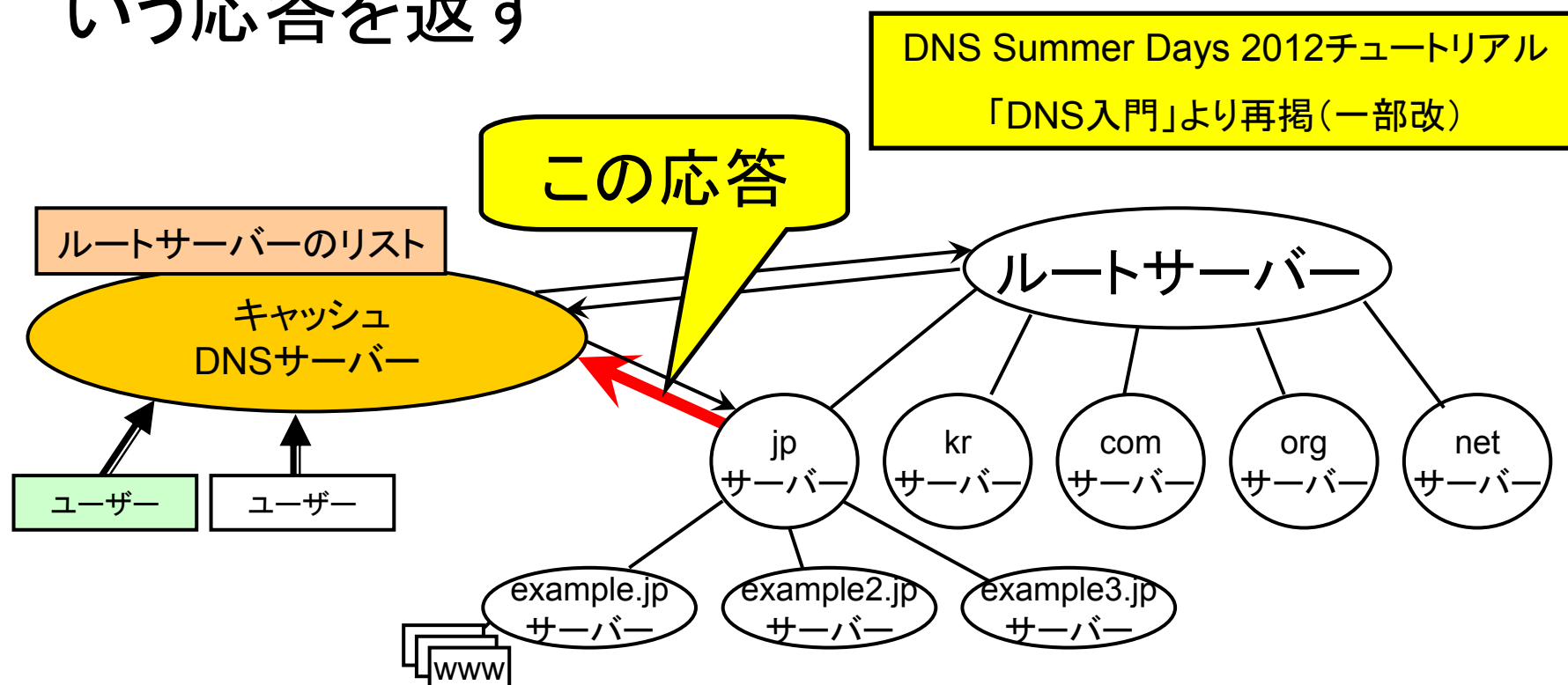
出力例に関する注意

- 本資料ではdigコマンドの出力例などにおいて実在するドメイン名の例を紹介していますが、それらは実際のものとは異なる場合があります
 - 説明に支障がない範囲で一部加工している場合があります
 - 実際の動作が原稿執筆時とは異なっている可能性があります

1. グループと内部名・外部名

よくある説明

- jpのサーバーはルートサーバーと同様「example.jpのサーバーに委任しています」という応答を返す



「委任しています」の内容

- digコマンドで確認してみる
 - 権威DNSサーバーへの問い合わせでは必ず+norecをつける
 - 最近のdig (BIND 9.9.0以降)では+noednsをつけないと、additional sectionの数が一つ多くなるので注意
 - EDNS0のOPT疑似レコードが付加されるため
- 典型的な出力例

```
$ dig +norec +noedns www.example.jp a @a.dns.jp
...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
example.jp.          86400   IN      NS      ns1.example.jp.

;; ADDITIONAL SECTION:
ns1.example.jp.     86400   IN      A       192.0.2.1
```

委任を示す応答

- 権威を持たない応答で、answer sectionが空
 - 権威を持たない: flagsにaaが含まれていない
 - answer sectionが空: answer sectionの数が0
- 応答が他のサーバーへの委任であることを示す

```
$ dig +nored +noedns www.example.jp a @a.dns.jp
...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
example.jp.          86400   IN      NS      ns1.example.jp.

;; ADDITIONAL SECTION:
ns1.example.jp.     86400   IN      A       192.0.2.1
```

委任情報とグループ

- authority sectionに委任先のNSホスト名を返す
- additional sectionに必要に応じて、NSホスト名のIPアドレスを返す
 - この場合に返されるA/AAAAをグループと呼ぶ

```
$ dig +nored +noedns www.example.jp a @a.dns.jp
...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
example.jp.          86400   IN      NS      ns1.example.jp.

;; ADDITIONAL SECTION:
ns1.example.jp.     86400   IN      A       192.0.2.1
```

グループと「グループ」

- しかし、グループに**相当する位置**にセットされる A/A AAA (以下「グループ」と表記)には、**受け入れてよい場合と受け入れてはいけない場合がある**
- 以降、本資料では、
 - 「グループ」: 委任を示す応答の additional section に返される NS ホスト名の A/A AAA
 - グループ: 「グループ」のうち、authority section で指定されている NS が **内部名** であるものの意味で使用します

問: この「グループ」は受け入れてよいか？

考えてみましょう

```
$ dig +nored +noedns www.example.jp a @a.dns.jp
...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
example.jp.          86400   IN      NS      ns1.example.net.

;; ADDITIONAL SECTION:
ns1.example.net.    86400   IN      A       192.0.2.1
```

答：受け入れてはいけない

- 委任先NSが**外部名**であった場合、「グループ」を受け入れてはいけない
- では、外部名であった場合、なぜ受け入れてはいけないのか？

```
$ dig +nored +noedns www.example.jp a @a.dns.jp
...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
example.jp.          86400   IN      NS      ns1.example.net.

;; ADDITIONAL SECTION:
ns1.example.net.    86400   IN      A       192.0.2.1
```


よくある説明(不十分:私もよくする)

- ns1.example.net→192.0.2.1という対応について、a.dns.jpは**権威を持っていない**
- つまり、実際には異なっているかもしれない
 - かつて実際に攻撃に使用(alternic.netへの誘導)

```
$ dig +noredns +noedns www.example.jp a @a.dns.jp
...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
example.jp.          86400   IN      NS      ns1.example.net.

;; ADDITIONAL SECTION:
ns1.example.net.    86400   IN      A       192.0.2.1
```

でも・・・

- 「権威を持っていない」ということ自体は、**委任先が内部名である場合も同様**
 - flagsにaa(権威を持つ応答)はセットされていない！
- では、内部名と外部名では**何が**違うのか？

```
$ dig +nored +noedns www.example.jp a @a.dns.jp
...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
example.jp.          86400   IN      NS      ns1.example.jp.
;; ADDITIONAL SECTION:
ns1.example.jp.     86400   IN      A       192.0.2.1
```

そもそも内部名とは何か？

- 委任先として指定したNSの名前が、委任先のゾーンの内側の名前だった場合、内部名 (in-bailiwick name) と呼ばれる
 - 親 (委任元) ではなく、委任先のゾーンで考える
- 内部名の例

```
$ORIGIN jp.  
@           86400 IN SOA ...  
example.jp. 86400 IN NS ns1.example.jp.
```

このゾーンの内側の名前

この名前が...

内部名についてのよくある誤解(1)

- × 委任元ゾーンの内部の名前はすべて内部名
– 内部名かどうかは委任元のゾーンではなく、**委任先**のゾーンによって決まる
- つまり、以下の例はいずれも内部名**ではない**

```
$ORIGIN jp.  
@           86400 IN SOA ...  
example.co.jp. 86400 IN NS ns1.example.ne.jp.
```

```
$ORIGIN jp.  
@           86400 IN SOA ...  
example.jp. 86400 IN NS ns1.example.ne.jp.
```

内部名についてのよくある誤解(2)

- × ルートゾーン内の委任はすべて内部名扱いになる
 - この誤解をしている人はかなり多いです
 - 実際の応答が「**グルー**」を返すことも理由の一つかも
- ルートゾーンであっても内部名・外部名の区別は、**他のゾーンと同様**(下記の例を参照)

```
$ORIGIN .  
@           86400 IN SOA ...                (内部名)  
jp.        86400 IN NS a.dns.jp.
```

```
$ORIGIN .  
@           86400 IN SOA ...                (外部名)  
com.       86400 IN NS a.gtld-servers.net.
```

問：これらは内部名か外部名か？

考えてみましょう

```

$ORIGIN jp.
@           86400 IN SOA ...
(1) example.jp.      86400 IN NS ns1.example.jp.
(2) example.jp.      86400 IN NS a.ns1.example.jp.
(3) example2.jp.     86400 IN NS ns1.example.com.
(4) example3.jp.     86400 IN NS ns1.example.jp.
(5) example.co.jp.   86400 IN NS ns1.example.co.jp.
(6) example.ne.jp.   86400 IN NS ns1.example.ad.jp.
(7) example4.jp.     86400 IN NS ns1.example.or.jp.

```

答：以下の通り

- 内部名：(1) (2) (5)
- 外部名：(3) (4) (6) (7)

```
$ORIGIN jp.  
@           86400 IN SOA ...  
(1) example.jp.      86400 IN NS ns1.example.jp.  
(2) example.jp.      86400 IN NS a.ns1.example.jp.  
(3) example2.jp.      86400 IN NS ns1.example.com.  
(4) example3.jp.      86400 IN NS ns1.example.jp.  
(5) example.co.jp.    86400 IN NS ns1.example.co.jp.  
(6) example.ne.jp.    86400 IN NS ns1.example.ad.jp.  
(7) example4.jp.      86400 IN NS ns1.example.or.jp.
```

無条件にグルーを受け入れてよいのは 内部名の場合のみ

- 委任先のNSが内部名であった場合、委任先のゾーンまたはその内側のゾーンが、NSとして指定された名前に対する権威を必ず持っている
- そのため、受け取ったグルーを一時的に信用してゾーンをたどっていけば、いずれはその名前に対し権威を持つ(本当に信用できる)権威DNSサーバーに必ず出会える
- そのため、名前解決において無条件にグルーを受け入れてよい

つまり、逆の立場から言うと...

- 委任先のNSが外部名であった場合、委任先でそのNSの名前に対し権威を持つ権威DNSサーバーに出会えるとは限らない
 - 偶然出会えることもある
 - 委任先がたまたまその外部名も管理していた場合
- そのため、その場合の「グループ」は受け入れてはいけない
 - 受け入れてよい場合もあるが、その判定は難しい

「グループ」が付く条件

- 最近の権威DNSサーバー(BIND 9、NSDなど)の実装では、外部名の「グループ」は**できるだけ付けない**ように動作する
 - 送られても「心ある」キャッシュは捨てる
 - つまり、**無駄なデータ**となる
- ただし、実装によっては「その権威DNSサーバー内において、**他の委任情報からグループとして参照されているA/AAAAは外部名であっても「グループ」として付ける**」という動作をするので注意

「グループ」の例

- www.iij.ad.jp
- www.iij-ii.co.jp
- www.iij4u.or.jp

...の、それぞれのAをJP DNSサーバーに問い合わせた際の、**現状における「グループ」の付き方**を見てみましょう

「グループ」の例(続き)

- www.iij.ad.jpの例
- 内部名なのでグループが付く

```
$ dig +nored +noedns www.iij.ad.jp a @a.dns.jp
...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 4

;; QUESTION SECTION:
;www.iij.ad.jp.                IN      A

;; AUTHORITY SECTION:
iij.ad.jp.                    86400  IN      NS      dns0.iij.ad.jp.
iij.ad.jp.                    86400  IN      NS      dns1.iij.ad.jp.

;; ADDITIONAL SECTION:
dns0.iij.ad.jp.              86400  IN      A       210.138.174.16
dns0.iij.ad.jp.              86400  IN      AAAA    2001:240:bb41:8002::1:16
dns1.iij.ad.jp.              86400  IN      A       210.138.175.5
dns1.iij.ad.jp.              86400  IN      AAAA    2001:240:bb4c:8000::1:5
```

「グループ」の例(続き)

- www.iij-ii.co.jpの例
- 外部名なのでグループは付かない

```
$ dig +nored +noedns www.iij-ii.co.jp a @a.dns.jp
...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 0

;; QUESTION SECTION:
;www.iij-ii.co.jp.                IN      A

;; AUTHORITY SECTION:
iij-ii.co.jp.                    86400  IN      NS      dns-b.iij.ad.jp.
iij-ii.co.jp.                    86400  IN      NS      dns-c.iij.ad.jp.
```

「グループ」の例(続き)

- www.iij4u.or.jpの例
- 外部名だが、iij.ad.jpから参照されているため「グループ」が付く

```
$ dig +nored +noedns www.iij4u.or.jp a @a.dns.jp
...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 4

;; QUESTION SECTION:
;www.iij4u.or.jp.                IN      A

;; AUTHORITY SECTION:
iij4u.or.jp.                    86400  IN      NS      dns0.iij.ad.jp.
iij4u.or.jp.                    86400  IN      NS      dns1.iij.ad.jp.

;; ADDITIONAL SECTION:
dns0.iij.ad.jp.                 86400  IN      A       210.138.174.16
dns0.iij.ad.jp.                 86400  IN      AAAA    2001:240:bb41:8002::1:16
dns1.iij.ad.jp.                 86400  IN      A       210.138.175.5
dns1.iij.ad.jp.                 86400  IN      AAAA    2001:240:bb4c:8000::1:5
```

この動作の理由(私の推測)

- その(委任情報として指定された)名前が内部名として扱われていることを、私(委任元の権威DNSサーバー)は**知っている**
- であれば、私がそれを「グループ」として返しても実害はない**はず**である(**安全性を損なわない**)
- そして、それらを「グループ」として返すことにより、名前解決の効率が良くなりうる
 - 実際にそうしたケースは起こりうる

それなら、「グループ」を**付けてあげる**ことにしましょう

余計なお世話？

- しかし、この動作はグループと内部名・外部名に対する**理解を妨げる**原因となっているとも言える
- かつ、この動作(DNSプロトコルとして明確に定義されていない)に依存したNS設定が出回ることにともなり、**将来の禍根**となる可能性がある
 - 実装を変更した際に名前が引けなくなるなど
- つまり「**余計なお世話**」であるとも言える

このパートのまとめ(1/2)

- グルーと「グルー」の違い
 - 「グルー」(このチュートリアル独自の定義):
委任を示す応答のadditional sectionに返されるNS
ホスト名のA/AAAA
 - グルー(一般的な定義):
「グルー」のうち、authority sectionで指定されている
NSが内部名であるもの
 - 「グルー」のうちグルーでないものを「**グルーもどき**」
(pseudo-glue)と呼ぶ流儀もある(一般的な定義で
はないので注意)

このパートのまとめ(2/2)

- 内部名の定義
 - 委任先として指定したNSの名前が、**委任先のゾーンの内側**の名前だった場合
 - 無条件にグループを受け入れていいのは、内部名の時だけ
- 「グループもどき」を付ける権威DNSサーバー
 - 権威DNSサーバーの実装によっては条件により、外部名の場合であっても「**グループもどき**」を付けることがあるので要注意

2. 委任情報とreferral

委任情報とreferral

- referral(参照)とは
 - 問い合わせに対し他のサーバーを案内すること
- 委任情報とreferral
 - 権威DNSサーバーはreferralを用いて委任を示す
 - 例「example.jpゾーンはns1.example.jpに委任しています」

```
$ dig +nored +noedns www.example.jp a @a.dns.jp
...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
example.jp.          86400   IN      NS      ns1.example.jp.

;; ADDITIONAL SECTION:
ns1.example.jp.     86400   IN      A       192.0.2.1
```

管理外のゾーンに対する問い合わせ

- その権威DNSサーバーの**管理外の名前**を問い合わせた場合の動作が問題となる
 - 自身または自身の委任先**ではない**名前
 - 例: JP DNSサーバーに対するns1.example.com
- 通常の名前解決では**実施されない**

「Upward referralsは有害」 (Upward Referrals Considered Harmful)

- DNS-OARCが2009年に公開した文書

Upward Referrals Considered Harmful

<<https://www.dns-oarc.net/oarc/articles/upward-referrals-considered-harmful>>

- NANOG 45の発表資料

“Upward Referrals Considered Harmful” – Peter Losher

<http://www.nanog.org/meetings/nanog45/presentations/Wednesday/Losher_light_harmful_N45.pdf>

Upward referralsとは？（続き）

- BIND 9では再帰検索要求の受け付けを無効にしている (recursion no;) サーバーに管理外の名前を問い合わせた場合、**ルートサーバーへのreferral**を返す
 - BIND 8では「**グルー**」も返す
- Upward referralsの例

```
$ dig +nored +noedns www.example.jp a @とある権威DNSサーバー
...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 0

;; AUTHORITY SECTION:
.           518400 IN      NS       a.root-servers.net.
.           518400 IN      NS       b.root-servers.net.
...
```

Upward referralsとは？（続き）

- この応答（上位ゾーンへのreferral）は、委任と同じ形をしているが、**委任ではない**
- この応答の意味（私の推測：**ツンデレ**風に）
 - 「これ、私が管理していないゾーンへの問い合わせね。だからほんとは無視してもいいんだから。」
 - 「でも特別に教えてあげる。あなたはルートサーバーに聞けばいいの。サーバーの一覧はこれね。」
 - 「わかった？ せっかく親切に教えてあげたんだから、もうここに来ちゃだめよ。」

この応答の根拠

- RFC 1034における以下の記述が、この応答の根拠となっていると考えられる

4.3.1. Queries and responses

...

The way that the name server answers the query depends upon whether it is operating in recursive mode or not:

- The simplest mode for the server is non-recursive, since it can answer queries using only local information: the response contains an error, the answer, or a referral to some other server “closer” to the answer. (応答はエラー、応答、あるいは「より近い」サーバーへのreferralを含む)

今となっては余計なお世話

- Upward referralsを返されても、キャッシュDNSサーバーは捨てるしかない
 - 委任情報ではない
 - ルートサーバーの一覧はそもそも持っている
- かつ、DNSリフレクター攻撃の元ネタになりうる
 - “. IN NS”の問い合わせ: 47バイト
 - その応答 (Upward referrals): 256バイト
- 実際の攻撃例あり
 - 米国ホスティング企業ISPrime社
 - “Upward Referrals Considered Harmful”にある実例

Upward referralsを出さない設定

- BIND 9
 - additional-from-cache no;を指定、または allow-query-cache { none; };を指定
 - BIND 9はこのデータをルートヒントとして持っている
 - ルートヒントはプライミングにより入手したキャッシュデータ
 - named.rootを消しただけでは効果がないので注意
 - 内部にハードコーディングされている
 - この設定により、REFUSEDを返すようになる

```
$ dig +norec +noedns www.example.jp a @とある権威DNSサーバー
...
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: xxxxx
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0
```

Upward referralsを出さない設定

- NSD: 設定不要
 - 管理外のゾーンに対する問い合わせには、デフォルトでSERVFAILを返す
- PowerDNS: 設定不要
 - デフォルトでsend-root-referral=noとなっている
 - yes (BIND 8互換) やlean (BIND 9互換) に設定可能
- djbdns: 設定不要
 - 管理外のゾーンに対する問い合わせには応答しない

余談：なぜルートヒントが？

- 権威DNSサーバー専用のBIND 9のはずなのに、なぜルートヒントが必要になるのか？
- 答：NOTIFYメッセージの送信先を調べている
 - BIND 9のプライマリサーバーは、ゾーンファイルの更新時にNOTIFYメッセージをセカンダリサーバー群に送信する
- この「セカンダリサーバー群」のIPアドレスを、プライマリサーバーのBIND 9はどうやって調べているのか？

余談:なぜルートヒントが?(続き)

- BIND 9のプライマリサーバーはnamed.confでNOTIFYの送り先を別途指定していない場合、以下のように動作する

- ①自分が管理するゾーンのNS RRSetを調べる
- ②①から、SOAのMNAMEにあるNSを削除する
- ③②で得られた名前を必要に応じて自分で名前解決し、NOTIFYを送る

- つまり、権威DNSサーバー専用にしたはずのBIND 9が、キャッシュDNSサーバーとして動作している!
- キャッシュDNSサーバー機能にのみ存在する脆弱性の影響を受ける可能性がある

BIND 9における NOTIFYのより安全な運用

- お奨め: この機構 (おせっかい) に頼らない
 - allow-transferと同様、named.confでIPアドレスを明示的に指定する
 - 外部のDNSサービスを利用する場合など、やむを得ない理由で外部名を使用する場合にも使える
 - 設定例は次ページを参照 (注: そのままコピー不可)
- 次善策: すべての権威DNSサーバーをそのゾー
ンの名前 (≠ 内部名) にする
 - 名前検索が必要なのは、権威を持たない情報のみ
 - 外部名や委任先の内部名であった場合、名前検索が必要になる

BIND 9における設定例

(注: NOTIFY関連のみの抜粋のためそのままコピー不可)

• プライマリサーバー

```
zone "example.jp" {  
    ....  
    // also-notifyで明示的(explicit)に指定したIPアドレスにのみNOTIFYを送る  
    notify explicit;  
    // NOTIFYを送るセカンダリサーバーのIPアドレス(変更されたらその都度更新)  
    also-notify { 192.0.2.53; 198.51.100.53; };  
};
```

• セカンダリサーバー

```
zone "example.jp" {  
    ....  
    // NOTIFYを送らない(指定しないと無駄なNOTIFYを送ろうとするので注意)  
    // notify no;でもNOTIFYの受信には影響しない  
    notify no;  
};
```


3. グルーはグルー (DNSデータのランキング)

重要なRFC: RFC 2181

- “Clarifications to the DNS Specification”
 - DNS仕様の明確化
- 5.4.1. Ranking data
 - データの出自 (source) が何に由来するかによって、そのデータの信頼度 (trustworthiness) が決まる
- 鍵となる要素
 - 権威を持つ応答か、持たない応答か (AAビットの有無)
 - answer、authority、additionalいずれのセクションにあったデータか (セクションの場所)
 - 権威を持つデータか、持たないデータか (応答内容)
 - CNAMEの処理に関係

RFC 2181 5.4.1. Ranking data (抜粋)

高い (most)

権威を持つ応答のanswer sectionの権威を持つデータ

権威を持つ応答のauthority sectionのデータ

権威を持たない応答のanswer sectionのデータ・
権威を持つ応答のanswer sectionの権威を持たないデータ

権威を持つ応答のadditional informationのデータ・
権威を持たない応答のauthority sectionのデータ・
権威を持たない応答のadditional informationのデータ



(注: 原文にadditional sectionではなくadditional informationと書かれている)

低い (least)

ここでのポイント

- 委任情報は「一番低い(the least)信頼度」
- 委任情報(NS)
 - 権威を持たない応答のauthority section
- グルー(A/A AAAA)
 - 権威を持たない応答のadditional information

グループ自身に対する問い合わせ

- 委任情報として答える必要あり
- BIND 9やNSDでは下記のように(正しく)答える
 - BIND 8にはグループ自身に対する問い合わせに対し、「権威を持たない応答のanswer section」として答えてしまう、という問題があった

```
$ dig +nored +noedns ns1.example.jp a @a.dns.jp
...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
example.jp.          86400   IN      NS      ns1.example.jp.

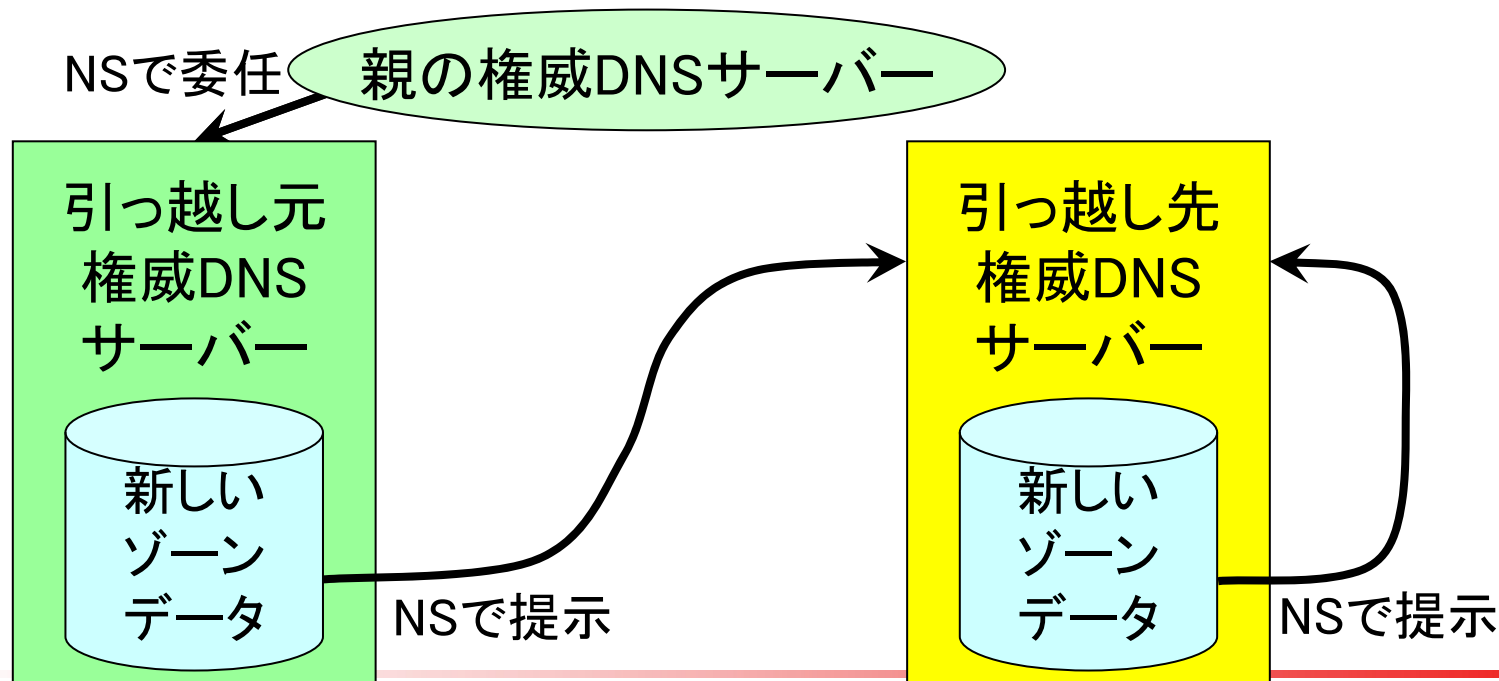
;; ADDITIONAL SECTION:
ns1.example.jp.     86400   IN      A       192.0.2.1
```

グループ自身に対する問い合わせ(続き)

- キャッシュDNSサーバーでは権威DNSサーバーから受け取った委任情報(NS)やグループを、**あくまで委任情報やグループとして扱わなければならない**
- つまり、クライアントからの問い合わせに対し、グループをanswer sectionに入れて答えてはならない
 - 信頼度の**増加(increase)**を招く
 - 通常の名前検索によってランキングがより高いデータを入手し、それを答える必要あり

例：権威DNSサーバーの引っ越し途中

- 親のNS/グループと子のNS/Aが異なっている
- 親のNS/グループは名前解決の途中で一時的に使われる(べき)
 - 子のNS/Aの方が親のNS/グループよりも信頼度が高い



再掲：内部名であった場合

- 委任先のNSが内部名であった場合、委任先のゾーンまたはサブゾーンが、NSとして指定された名前に対する権威を必ず持っている
- そのため、受け取ったグループを一時的に信用してゾーンをたどっていけば、いずれはその名前に対し権威を持つ（本当に信用できる）権威DNSサーバーに必ず出会える
- そのため、名前解決において無条件にグループを受け入れてよい

ランキングの管理はきわめて面倒

- 注意深く実装しないと、**バグ**や**脆弱性**のもとになりやすい
- 例：親のNS/グループと子のNS/A
 - 子のNS/Aの方が親のNS/グループよりも信頼度が上
 - 優先的に取り扱われなければならない
 - しかし、子の信頼度が高いのは子が子である間だけ
 - 信頼度が低い親のデータがなくなることで、子のデータがそもそも信用できなくなる(子が子でなくなる)
- 例：いわゆる浸透問題や幽霊ドメイン名脆弱性

そして、あの**有名な脆弱性**も...

4.カミングスキー型攻撃手法

カミンスキー型攻撃手法

- ダン・カミンスキー氏が2008年に指摘した攻撃手法
- しかし、カミンスキー氏が示したオリジナルの方法では、キャッシュポイズニングはできない
 - 説明は省略します(重要なことなので各自ご確認ください)
- **問題点を修正**した攻撃方法がいくつか発表されている
 - 中京大学の鈴木常彦先生が実験(*1)で使われた方法
 - 以下「**パターン1**」「**パターン2**」とします
 - 「実践DNS」に記述した方法
 - 以下「**パターン3**」とします

(*1) BIND各バージョンへの毒入れ実験 (Kaminsky Bug の検証)
<<http://www.e-ontap.com/dns/bindmatrix.html>>

カミンスキー型攻撃手法(パターン1)

- 鈴木先生のWebページの方法をもとに作成

```
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;(random).example.jp. IN      A

;; ANSWER SECTION:
(random).example.jp.      86400    IN      A      (適当なIPアドレス)

;; AUTHORITY SECTION:
(random).example.jp.      86400    IN      NS      www.example.jp.

;; ADDITIONAL SECTION:
www.example.jp.           86400    IN      A      192.0.2.234
(キャッシュさせたい偽情報)
```

カミンスキー型攻撃手法(パターン2)

- 鈴木先生のWebページの方法をもとに作成

```
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;(random).example.jp. IN      A

;; ANSWER SECTION:
(random).example.jp.      86400    IN      A      (適当なIPアドレス)

;; AUTHORITY SECTION:
example.jp.               86400    IN      NS      www.example.jp.

;; ADDITIONAL SECTION:
www.example.jp.          86400    IN      A      192.0.2.234
(キャッシュさせたい偽情報)
```

カミンスキー型攻撃手法(パターン3)

- 「実践DNS」p.298より

```
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;(random).www.example.jp. IN      A

;; ANSWER SECTION:
(random).www.example.jp. 86400   IN      A      (適当なIPアドレス)

;; AUTHORITY SECTION:
www.example.jp.          86400   IN      NS      www.example.jp.

;; ADDITIONAL SECTION:
www.example.jp.          86400   IN      A      192.0.2.234
(キャッシュさせたい偽情報)
```

それぞれの違い

- 「パターン1」と「パターン2」の違い
 - authority sectionの内容
 - パターン1: (random).example.jpゾーンのNS
 - パターン2: example.jpゾーンのNS
- 「パターン1」「パターン2」と「パターン3」の違い
 - 問い合わせ/answer sectionの内容
 - パターン1/2: (random).example.jpのA
 - パターン3: (random).**www.example.jp**のA
 - authority sectionの内容
 - パターン1/2: (上記参照)
 - パターン3: **www.example.jp**ゾーンのNS

ポイント

- ランキングの定義により、「パターン1」「パターン2」の「権威を持つ応答のadditional informationのデータ」だけでは、**キャッシュポイズニングは成立しない(はず)**
 - 鈴木先生の実験の通り
- しかし、それが**成立してしまう**BIND 9が存在した
- そして、その**バグ**は下記により修正された

2786. [bug] Additional could be promoted to answer. [RT #20663]
([bug] Additionalがanswerに上昇する場合があります)

ポイント(続き)

- 「パターン3」では、
 - 権威を持つ応答のauthority sectionのデータ
 - 権威を持つ応答のadditional informationのデータの**双方**に、www.example.jp (**標的名**)を指定している
- これにより、攻撃が可能となる
 - ただし、**追加の工夫**が必要(後述)
- ただし、この時点で**通常の名前検索**で得た、
 - 権威を持つ応答のanswer sectionの権威を持つデータを**既にキャッシュしていた**場合、**攻撃は失敗する(はず)**
 - このことは鈴木先生の実験でも説明されている

追加の工夫

- そして、キャッシュさせたい偽情報の、
 - 権威を持つ応答のauthority sectionのデータ
 - 権威を持つ応答のadditional informationのデータ
 をもとに、
 - 権威を持つ応答のanswer sectionの権威を持つデータをキャッシュさせるための工夫が必要になる
- 「実践DNS」では脚注に記述

3 攻撃者が攻撃を成功させるためには、192.0.2.234でwww.example.jpの偽の権威DNSサーバーを立ち上げ、グルーと同じ偽の情報を設定しておく必要がある。

4 キャッシュDNSサーバーが問い合わせに用いるUDPポートが固定であった場合、数秒以内にDNSキャッシュポイズニングを成立できる。

5 本書執筆時点では他の脆弱性やDNSSECへの対応などを含め、9.7.3以降のバージョンを使うことが推奨されている。

6 計算上、ポートを固定していた場合に比して攻撃の成功率を65536分の1まで低減できる。

7 インターネットに対して、送信元アドレスを偽造したパケットを送出できなくする技術。特に、ISPなどでの対策となる。

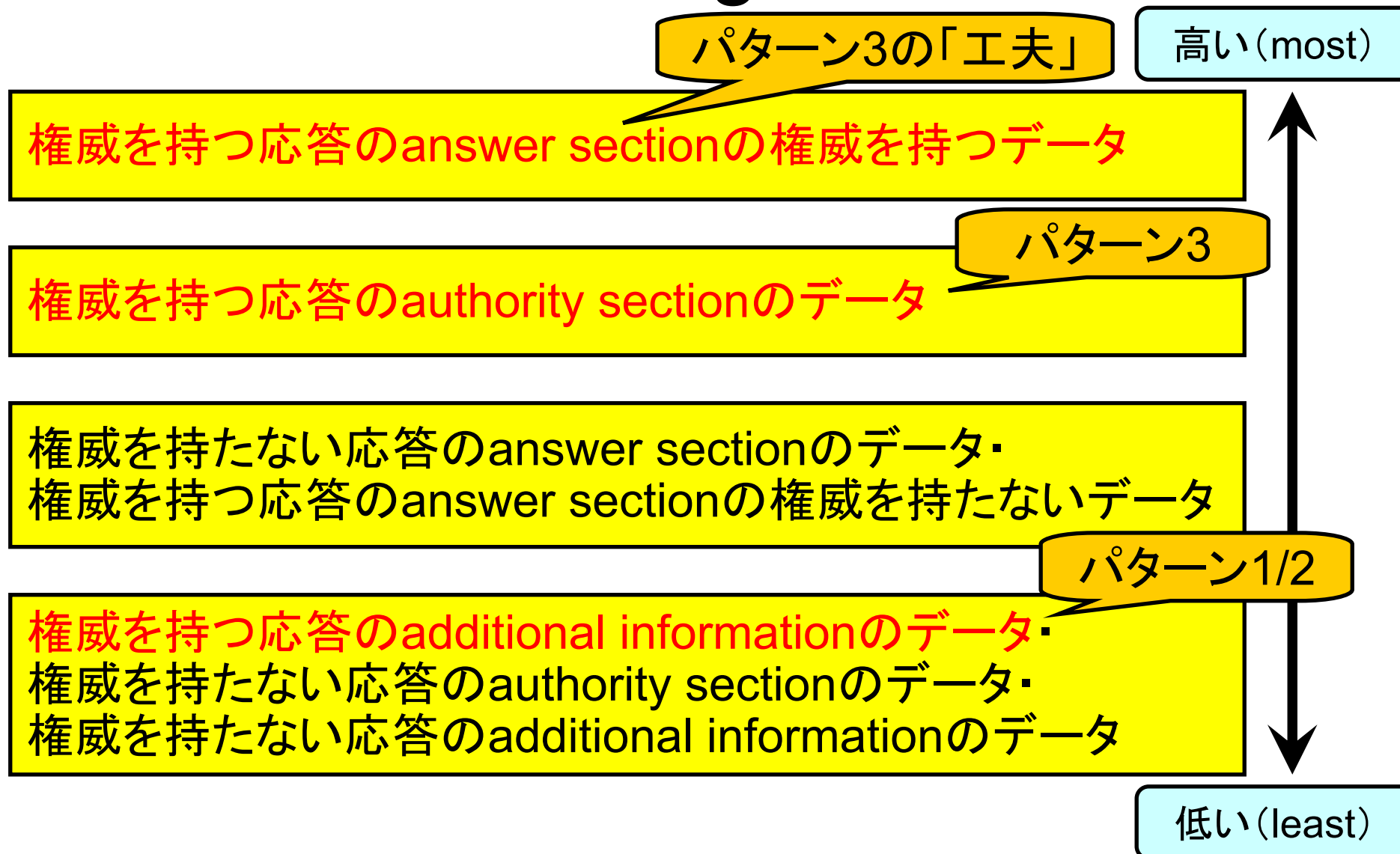
299

追加の工夫によるランキングの上昇

- 追加の工夫: 偽の委任先(この例では192.0.2.234)で偽の権威DNSサーバーを立ち上げ、キャッシュさせたい偽情報と同じデータを設定しておく
- キャッシュDNSサーバーが、攻撃者が設定した偽の権威DNSサーバー上の偽情報を参照した時点で「権威を持つ応答のanswer sectionの権威を持つデータ」としてキャッシュされ、キャッシュポイズニングが成功する

セキュリティ上の理由により、
具体的な方法の紹介は省略します

5.4.1. Ranking data (再掲)



まとめ:カミンスキー型攻撃手法

- 手法の技術的理解とその検証が**非常に面倒**
 - このパートの話は**難しかった**と思います(ごめんなさい)
- RFC 2181を読み込まなければわからない
 - 実のところ、読み込んででもよくわからない
 - 実は**私もわかっていない**のかも
- かつ、**実装上のバグ**も絡んでくる
 - 実装におけるランキングの管理はきわめて面倒
- そして、**RFC 2181のランキングそのもの**や**現存する実装**が**完全なものかどうか**は**誰にもわからない**

Q&A

