DNS完全解説 の仕組み

第3回 逆引きの仕組みと逆引き用ゾーンデータファイルの記述法

馬場達也

前回は、権威ネームサーバの持つ正引き用のゾーンデータファイルの内容とその記述法を解説した。第3回目となる今回は、逆引きの仕組みと、逆引き用のゾーンデータファイルの記述法について紹介しよう。

IPアドレスから ホスト名を得る逆引き

DNSでの名前解決は、ホスト名からIPアドレスを得ることを目的としている。しかし、IPアドレスからホスト名を得るためにDNSを使用することもできる。ホスト名からIPアドレスを得ることを「正引き」と呼び、IPアドレスからホスト名を得ることを「逆引き」と呼ぶ。

では、逆引きはどのような場合に利用されるのだろうか。逆引きは、主にサーバ側で、アクセスしてきたホストのホスト名をログに残すために使用される。アクセスしてきたホストのIPアドレスは、アクセスに使用されたパケットの送信元IPアドレスから判明するが、IPアドレスだけではどの組織のホストがアクセスしてきたのかがわからない。このため、IPアドレスを逆引きし、ドメイン名を含むホスト名に変換してログに残すのである。

TelnetやFTP、SSHなどのサーバで逆引きを行っている場合には、逆引きが設定されていないと、逆引き問い合わせのタイムアウト待ちが発生し、ログインするまでに時間がかかったりする。また、場合によっては、ホストの身元が不明であるとしてサーバが利用を拒否することもある。このため、正引きと同様に、逆引きも正しく設定しておくことが重要である。

PTRレコードを使用して IPアドレスからホスト名に変換

正引きでは、Aレコードを使用して、ホスト名からIPアドレスへの変換を行うが、逆引きでは、PTR（Pointer）レコードというリソースレコードを使用して、IPアドレスからホスト名への変換を行う。このPTRレコードは、RFC 1035で規定されており、次のような書式になっている。

```
<owner> <ttl> <class> PTR <dname>
```

<owner>には、ドメイン名形式に変換されたIPアドレスを記述し、<dname>には、そのIPアドレスに該当するホスト名をFQDN（Fully Qualified Domain Name）で記述する。<ttl>には、このリソースレコードのキャッシュの有効期間を秒単位で記述する。<ttl>を省略した場合には、直前の$TTL制御ステートメントでセットされたデフォルトの有効期間がセットされる。<class>には、ネットワーククラスを記述し、インターネットでは「IN」と記述する。<class>を省略した場合には、自動的に「IN」がセットされる。

<owner>には、本来はIPアドレスを記述するべきであるが、IPアドレスは「192.168.0.10」のように記述され、同じ組織のホストでは、前半部分（「192.168.0」の部分）が固定され、後半部分（最後の「10」の部分）が変化する。これは前半部分が変化し、後半部分が固定となるドメイン名とはまったく逆の性質である。このため、IPアドレスをドメイン名と同じように扱えるようにするために、PTRレコードの<owner>には、IPアドレスを左右逆に記述し、最後に「in-addr.arpa.」というドメイン名を付加したものを記述する。例えば、IPアドレスが「192.168.0.10」の場合は、「10.0.168.192. in-addr.arpa.」のようになる。

逆引き用のDNSツリーと in-addr.arpaドメイン

連載第1回では、DNSは、ルートネームサーバを頂点としたツリー構造で管理されていることを説明した。逆引きも同じDNSツリーで管理され、そのためのドメインとしてin-addr.arpaドメインが割り当てられている（図1）。

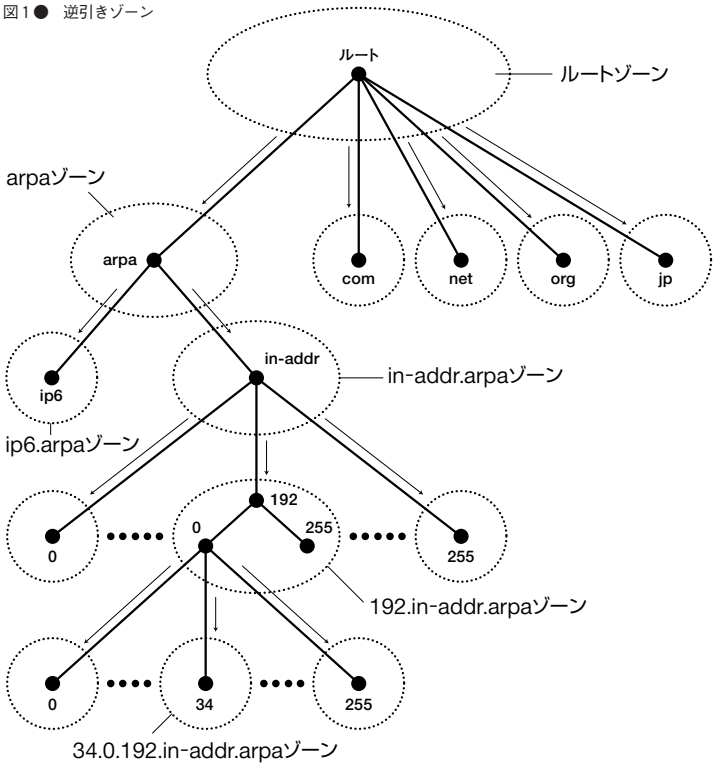
arpaドメインは、かつては、インターネットの前身

であるARPANETの時代に使用されていたホストの名称を現在のDNSに移行させるために使用されていたが、現在は“Address and Routing Parameters Area”の頭字語として定義し直され、アドレスなどを管理する特殊なドメインとして利用されている（arpaドメインについての詳細は、RFC 3172を参照のこと）。arpaドメインには、IPv4アドレス用の逆引きドメインであるin-addr.arpaドメインのほかに、IPv6アドレス用の逆引きドメインであるip6.arpaドメイン（RFC 3152で規定）や、IP電話用の番号を管理するためのドメインであるe164.arpaドメイン（RFC 2916で規定）がある（e164.arpaは暫定なので今後変更される可能性がある）。

in-addr.arpaドメインは、IPアドレスの割り当てを行う地域インターネットレジストリの1つであるARIN（American Registry for Internet Numbers）が管理し、あるアドレスブロックが割り当てられている組織に、その部分の逆引きゾーンの管理を委任している。例えば、アドレスブロック「192.0.34.0/24」はICANN（The Internet Corporation for Assigned Names and Numbers）に割り当てられているが、ARINは、このアドレスに対する逆引きゾーンである「34.0.192.in-addr.arpaゾーン」の管理をICANNに委任している（図中のin-addr.arpaゾーンや192.in-addr.arpaゾーンは、ARINが管理している）。

それでは、192.0.34.72というアドレスの逆引きを行う場合を図2を例に説明しよう。基本的には、正引きの場合と同様である。まず、クライアントのリゾルバ

図1● 逆引きゾーン



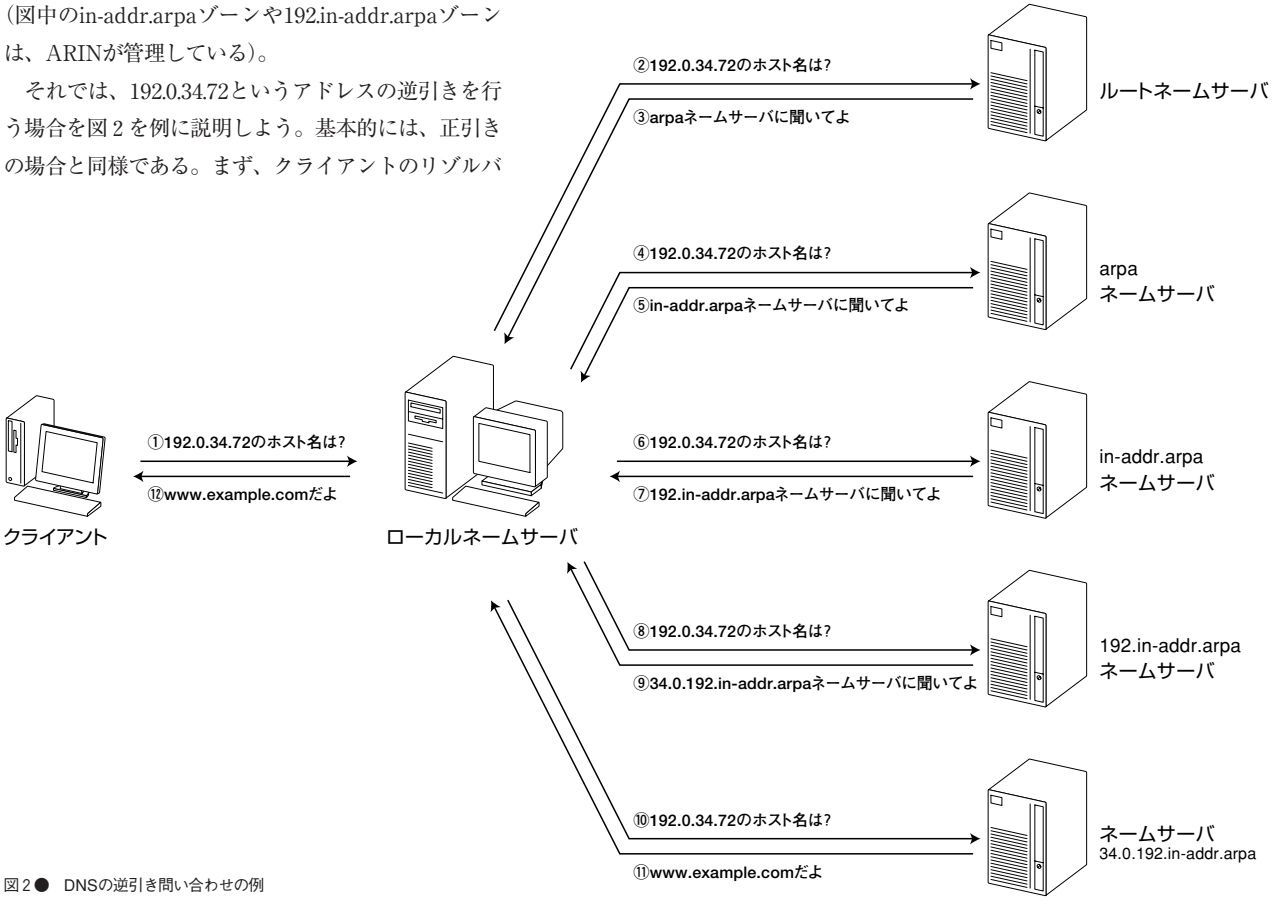


図2● DNSの逆引き問い合わせの例

が、あらかじめ設定された近くのローカルネームサーバに対して、逆引きの問い合わせを行う(①)。逆引きでは、実際は「72.34.0.192.in-addr.arpa.に対するPTRレコードの内容」というように問い合わせる。もし、このローカルネームサーバに逆引きの結果がキャッシュとして残っていれば、その結果がクライアントのリゾルバに返され、問い合わせが完了する。

問い合わせ内容がキャッシュされていない場合は、ローカルネームサーバは、最初にルートネームサーバに対して問い合わせを送る(②)。すると、ルートネームサーバは、arpaゾーンを管理しているネームサーバのホスト名とIPアドレスを返してくるので(③)、arpaゾーンを管理しているネームサーバに対して、再び逆引きの問い合わせを行う(④)。すると、arpaゾーンを管理しているネームサーバは、in-addr.arpaゾーンを管理しているネームサーバのホスト名とIPアドレスを返してくるので(⑤)、in-addr.arpaゾーンを管理しているネームサーバに対して、再び逆引きの問い合わせを行う(⑥)。

in-addr.arpaゾーンを管理しているネームサーバは、これに応じて192.in-addr.arpaゾーンを管理しているネームサーバのホスト名とIPアドレスを返してくるので(⑦)、192.in-addr.arpaゾーンを管理しているネームサーバに対して、再び逆引きの問い合わせを行う(⑧)。すると、192.in-addr.arpaゾーンを管理しているネームサーバは、34.0.192.in-addr.arpaゾーンを管理しているネームサーバのホスト名とIPアドレスを返してくるので(⑨)、34.0.192.in-addr.arpaゾーンを管理しているネームサーバに対して、再び逆引きの問い合わせを行う(⑩)。34.0.192.in-addr.arpaゾーンを管理しているネームサーバは、72.34.0.192.in-addr.arpa.に対するPTRレコードを管理しているため、その内容を返答する(⑪)。ローカルネームサーバは、判明したPTRレコードの内容を、問い合わせ元であるクライアントのリゾルバに返答する(⑫)。

逆引き用のゾーンデータファイルを記述する

逆引きを行えるようにするためには、前回紹介した正引き用のゾーンデータファイルとは別に、逆引き用のゾーンデータファイルを作成して管理する必要がある。リスト1に、192.168.0.0/24というアドレスを使用している組織における逆引き用ゾーンデータファイル（0.168.192.in-addr.arpaゾーン）の記述例を示す。正引き用ゾーンデータファイルの場合と同様に、起点ドメイン名（ownerには「@」と記述される）に対しては、必ずSOAレコードとNSレコードを記述する。そのあとに、各IPアドレスに対してPTRレコードを記述していけばよい。

逆引き用ゾーンデータファイルを記述する際の注意事項

ここでは、逆引き用ゾーンデータファイルを記述する際の注意事項について説明する。ここで紹介する注意事項については、RFC 1912で詳しく述べられているので、こちらもぜひ参照していただきたい。

存在するホストに対しては、すべて逆引きできるように設定すること

サーバにアクセスする場合は、アクセス先のサーバが、アクセス元のIPアドレスからクライアントのホスト名を逆引きする場合があるので、サーバにアクセスする可能性のあるホストに対しては、すべて逆引き可能に設定しておく必要がある。特に、DHCP（Dynamic Host Configuration Protocol）でアドレスを自動的に

```
$TTL 86400
@     IN    SOA     ns1.example.com.hostmaster.example.com.(
                            2002081800  ;シリアル番号
                            28800       ;リフレッシュ間隔（秒）
                            7200        ;リトライ間隔（秒）
                            604800      ;ゾーンの有効期間（秒）
                            3600        ;ネガティブキャッシュの有効期間（秒）
                            )
      IN    NS      ns1.example.com.    ;このゾーンのプライマリマスタ
      IN    NS      ns2.example.com.    ;このゾーンのセカンダリマスタ

10    IN    PTR     ns1.example.com.
11    IN    PTR     ns2.example.com.
20    IN    PTR     mx1.example.com.
21    IN    PTR     mx2.example.com.
30    IN    PTR     www.example.com.
```

リスト1● 逆引き用ゾーンデータファイルの記述例（0.168.192.in-addr.arpaゾーン）

割り当てているホストの場合は、忘れてしまいがちであるが、正引きおよび逆引きの両方ができるようにしておく。このとき登録するホスト名は、「dhcp-192-168-0-52.example.com」のようなものでよい。ただし、NAT（Network Address Translation）によってIPアドレスを変換している場合は、変換後のIPアドレスで逆引きできるようにしておく必要がある。

Aレコードの内容とPTRレコードの内容を合わせること

逆引きを行うサーバの中には、逆引きして得られたホスト名をさらに正引きし、その結果得られたIPアドレスと、アクセスに利用された実際のIPアドレスを比較するものがある。ここで、両者のIPアドレスが異なった場合は、サーバがアクセスを拒否する場合があるので、PTRレコードの内容とAレコードの内容を正確に合わせておく必要がある。また、仕様では、CNAMEで設定した別名をPTRレコードに登録することはできないと規定されているので注意しよう。

【誤った記述例】

（正引き用ゾーンデータファイル）

```
$ORIGIN example.com.
www   IN    CNAME   hoge
hoge  IN    A   192.168.0.30
```

（逆引き用ゾーンデータファイル）

```
$ORIGIN 0.168.192.in-addr.arpa.
30    IN    PTR   www.example.com
                  ↑
```

「www」は正規名でなく別名である

【正しい例】

（正引き用ゾーンデータファイル）

```
$ORIGIN example.com.
www   IN    CNAME hoge
hoge  IN    A   192.168.0.30
```

（逆引き用ゾーンデータファイル）

```
$ORIGIN 0.168.192.in-addr.arpa.
30 IN PTR hoge.example.com
                ↑
```

Aレコードの内容と合わせる

同じアドレスに対して、Aレコードを複数記述しないこと

PTRレコードでは、あるIPアドレスに対して1つしか正規名を設定できない。このため、同じIPアドレスに対してAレコードを複数設定した場合は、PTRレコードで設定した正規名以外は逆引きできないことになってしまう。このため、同じIPアドレスに対して複数のホスト名を設定したい場合は、Aレコードを複数記述するのではなく、CNAMEレコードを使用する。

【誤った記述例】

（正引き用ゾーンデータファイル）

```
$ORIGIN example.com.
mail  IN A 192.168.0.30←
ns    IN A 192.168.0.30←
```

このAレコードに対する逆引きレコードがない

逆引きレコードが登録されている

（逆引き用ゾーンデータファイル）

```
$ORIGIN 0.168.192.in-addr.arpa.
30    IN PTR ns
```

【正しい例】

（正引き用ゾーンデータファイル）

```
$ORIGIN example.com.
mail  IN CNAME ns ←
ns    IN A 192.168.0.30
```

CNAMEレコードにする

（逆引き用ゾーンデータファイル）

```
$ORIGIN 0.168.192.in-addr.arpa.
30    IN PTR ns
```

同じホスト名に対してAレコードを複数記述した場合は、すべてのAレコードに対してPTRレコードを記述すること

負荷分散などを目的として、同じホスト名に対して複数のAレコードを記述している場合には、そのすべてのAレコードに対してPTRレコードを記述する。

【誤った記述例】

（正引き用ゾーンデータファイル）

```
$ORIGIN example.com.
www   IN   A 192.168.0.30
      IN   A 192.168.0.31
```

（逆引き用ゾーンデータファイル）

```
$ORIGIN 0.168.192.in-addr.arpa.
30    IN   PTR   www.example.com
↑
```

192.168.0.30に対するPTRレコードしか登録されていない

```
$ORIGIN 0.168.192.in-addr.arpa.
$TTL 86400
@     IN     SOA     ns1.example.com. hostmaster.example.com.(
                          2002081800 ;シリアル番号
                          28800      ;リフレッシュ間隔（秒）
                          7200       ;リトライ間隔（秒）
                          604800     ;ゾーンの有効期間（秒）
                          3600       ;ネガティブキャッシュの有効期間（秒）
                          )
      IN     NS      ns1.example.com.
      IN     NS      ns2.example.com.
;
;192.168.0.0/26用
;
0-26  IN     NS      ns1.example1.com.
      IN     NS      ns2.example1.com.
1     IN     CNAME   1.0-26.0.168.192.in-addr.arpa.
2     IN     CNAME   2.0-26.0.168.192.in-addr.arpa.
3     IN     CNAME   3.0-26.0.168.192.in-addr.arpa.
...
63    IN     CNAME   63.0-26.0.168.192.in-addr.arpa.
;
;192.168.0.64/26用
;
64-26IN      NS      ns1.example2.com.
      IN     NS      ns2.example2.com.
65    IN     CNAME   65.64-26.0.168.192.in-addr.arpa.
66    IN     CNAME   66.64-26.0.168.192.in-addr.arpa.
67    IN     CNAME   67.64-26.0.168.192.in-addr.arpa.
...
12    IN     CNAME   127.64-26.0.168.192.in-addr.arpa.
;
;192.168.0.128/26用
;
128-26      IN      NS    ns1.example3.com.
      IN     NS      ns2.example3.com.
129   IN     CNAME   129.128-26.0.168.192.in-addr.arpa.
130   IN     CNAME   130.128-26.0.168.192.in-addr.arpa.
131   IN     CNAME   131.128-26.0.168.192.in-addr.arpa.
...
191   IN     CNAME   191.128-26.0.168.192.in-addr.arpa.
;
;192.168.0.192/26用
;
192-26      IN      NS    ns1.example4.com.
      IN     NS      ns2.example4.com.
193   IN     CNAME   193.192-26.0.168.192.in-addr.arpa.
194   IN     CNAME   194.192-26.0.168.192.in-addr.arpa.
195   IN     CNAME   195.192-26.0.168.192.in-addr.arpa.
...
255   IN     CNAME   255.192-26.0.168.192.in-addr.arpa.
```

リスト2● CIDRに対応した逆引き用ゾーンデータファイルの記述例（0.168.192.in-addr.arpaゾーン）

```
$ORIGIN 0-64.0.168.192.in-addr.arpa.
$TTL 86400
@     IN     SOA   ns1.example1.com. hostmaster.example1.com.(
                        2002081800 ;シリアル番号
                        28800      ;リフレッシュ間隔（秒）
                        7200       ;リトライ間隔（秒）
                        604800     ;ゾーンの有効期間（秒）
                        3600       ;ネガティブキャッシュの有効期間（秒）
                        )
      IN     NS    ns1.example1.com.
      IN     NS    ns2.example1.com.

3     IN     PTR   hoge.example1.com.
10    IN     PTR   ns1.example1.com.
11    IN     PTR   ns2.example1.com.
20    IN     PTR   mx1.example1.com.
21    IN     PTR   mx2.example1.com.
30    IN     PTR   www.example1.com.
```

リスト3● CIDRに対応した逆引き用ゾーンデータファイルの記述例（0-64.0.168.192.in-addr.arpaゾーン）

【正しい例】

（正引き用ゾーンデータファイル）

```
$ORIGIN example.com.
www     IN      A       192.168.0.30
        IN      A       192.168.0.31
```

（逆引き用ゾーンデータファイル）

```
$ORIGIN 0.168.192.in-addr.arpa.
30      IN      PTR     www
31      IN      PTR     www
↑
192.168.0.31に対するPTRレコードも記述する
```

CIDRブロックでアドレスが割り当てられている場合の記述法

もし、アドレスブロックが8ビット単位（8ビットマスク、16ビットマスク、24ビットマスクのいずれか）で割り当てられているのであれば、逆引き用のゾーンの委任は正引きの場合と同様に行うことができる。しかし、CIDR（Classless Inter-Domain Routing）によって、アドレスブロックが26ビットマスクで割り当てられている場合はどうだろうか。この場合には、例えば、192.168.0.0/26、192.168.0.64/26、192.168.0.128/26、192.168.0.192/26というアドレスブロックがそれぞれ異なる組織に割り当てられる。しかし、これまでに説明した逆引きの仕組みでは、これらの4つのアドレスブロックは同じ「0.168.192.in-addr.arpaゾーン」に含まれるため、複数の組織で同じゾーンデータファイルを管理しなければならないことになり、非常に不便だ。

このため、アドレスブロックがCIDRで割り当てられている場合でも、逆引き用ゾーンデータファイルを分割して管理する方法が考案されており、RFC 2317に記述されている。この方法では、ネットワークごとに新たなサブドメインを作ることによって、ゾーンを分けるようにしている。

例として、26ビットマスクのアドレスブロック用の逆引き用ゾーンデータファイルを分割して管理する場合のゾーンデータファイルの内容をリスト2およびリスト3に示す。

リスト2の0.168.192.in-addr.arpaゾーンのゾーンデータファイルは上位組織が管理し、ここでネットワークごとに新たなサブドメインを作成する。このサブドメインはどのような文字列でもよいが、例えば「192.168.0.0/26」というネットワーク用のサブドメインであれば、ネットワークアドレスの最後の8ビットぶんの「0」とネットマスクの「26」を組み合わせて「0-26.0.168.192.in-addr.arpa.」というようなサブドメインを作成する。また、それに合わせて、*.0.168.192.in-addr.arpa.（*の部分は、1〜63の数字）というドメイン名を*.0-26.0.168.192.in-addr.arpa.というドメイン名に変換するためのCNAMEレコードを作成する。これを、他の192.168.0.64/26、192.168.0.128/26、192.168.0.192/26の各ネットワーク用にも作成する。そして、作成した「0-26.0.168.192.in-addr.arpaゾーン」「64-26.0.168.192.in-addr.arpaゾーン」「128-26.0.168.192.in-addr.arpaゾーン」「192-26.0.168.192.in-addr.arpaゾーン」を、それぞれ、該当するアドレスブロックを使用している組織が管理する（リスト3は、0-26.0.168.192.in-addr.arpaゾーンの例である）。

このような仕組みで本当にうまく逆引きができるのだろうか。例えば、「3.0.168.192.in-addr.arpa.のPTRレコード」を問い合わせる場合を考えてみよう。ルートネームサーバから順に問い合わせていくと、0.168.192.in-addr.arpaゾーンを管理しているネームサーバからは、「3.0.168.192.in-addr.arpa.の正規名は3.0-26.0.168.192.in-addr.arpa.である」という内容のCNAMEレコードと、「0-26.0.168.192.in-addr.arpaゾーン」を管理しているネームサーバのホスト名とIPアドレスが返される。そこで、0-26.0.168.192.in-addr.arpaゾーンを管理しているネームサーバに対して「3.0-26.0.168.192.in-addr.arpa.のPTRレコード」を問い合わせる。すると、リスト3にあるように、「hoge. example1.com」というホスト名が答えとして返されるのである。

次回は、メール配送におけるDNSの役割について説明する。

NTTデータ　馬場達也

●今回の内容に関連するRFC

RFC 1035 "Domain Names - Implementation and Specification"
RFC 1912 "Common DNS Operational and Configuration Errors"
RFC 2317 "Classless IN-ADDR.ARPA delegation"
RFC 2916 "E.164 number and DNS"
RFC 3152 "Delegation of IP6.ARPA"
RFC 3172 "Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain ("arpa")"