新連載

DNS設定駆け込み寺

勘所をしっかり押さえる

第1回

DNSは、インターネットでアプリケーションを使用する場合になくてはならない重要なシステムである。しかし、それだけに、DNSの運用については苦労している管理者も多いと思う。本連載では、実際にネームサーバ（DNSサーバ）を運用する際に生じるさまざまな疑問に答え、管理者がスムーズにDNSを運用するためのテクニックを紹介する。

Q11 BINDをリモートから制御するには？

ネームサーバを複数台管理している場合など、それぞれのサーバにわざわざログインして作業するのは非常に面倒だ。リモートからネームサーバを制御する方法はないのか？

A BIND 9に付属しているrndcを使用する

rndc（remote name daemon control）は、BIND 9に付属するリモート制御プログラムである。rndcはクライアントにインストールするプログラムで、クライアント側とサーバ側（named）であらかじめ共有された秘密鍵を使用して認証することにより、設定ファイルやゾーンデータのリロード、キャッシュデータのフラッシュなどをリモートから制御することを可能としている。

rndcの設定は、サーバ側はBINDの設定ファイルである「named.conf」ファイルで行い、クライアント側は「rndc.conf」ファイルで行う。

サーバ側ではcontrolsステートメントで設定する

サーバ側では、「/etc/named.conf」ファイルのcontrolsステートメントで設定を行う。controlsステートメントの書式は次のようになっている。

```
controls {
    inet <ip_addr> port <ip_port>
    allow { <address_match_list>; } keys { <key-name>; };
};
```

<ip_addr>と<ip_port>には、rndcの制御を受けるサーバ側のアドレスとポートを指定する。ネームサーバが複数のアドレスを持っていて、rndcからアクセスされるアドレスを限定したい場合には、<ip_addr>にそのアドレスを記述すればよいが、通常はワイルドカードを示す「*」と記述しておけばよいだろう。また、rndcは、デフォルトでTCPの953番ポートを使用するので、<ip_port>には「953」と指定しておく。また、<address_match_list>には、rndcによる制御を許可するクライアントのIPアドレスを記述する。そして、<key-name>には、使用する認証鍵の名称を記述する。この認証鍵は「/etc/rndc.key」ファイルで設定する。例えば、図1のように、ローカルホストと192.168.0.11のアドレスを持つクライアントからrndcによる制御を許可する場合には、リスト1のように記述する。

そして、リスト2のように、「/etc/rndc.key」ファイルで認証鍵を設定する。

この「/etc/rndc.key」ファイルは、namedを動作させるユーザー以外が内容

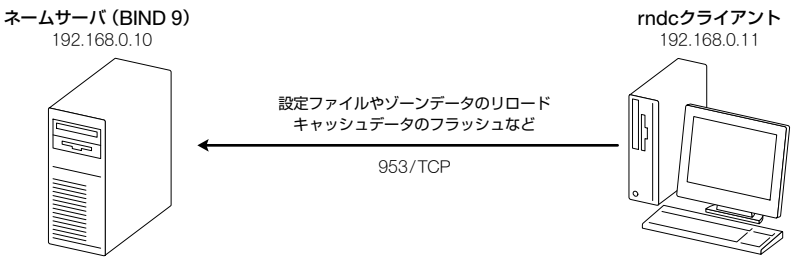

図1● BIND 9に付属するrndcを使えば、リモートからネームサーバを制御できる

```
controls {
        inet * port 953 ―――――――――――――――――――――――――― ポート953でrndcの制御を受ける
        allow { localhost; 192.168.0.11; } keys { rndc-key; }; ―― ローカルホストと192.168.0.11からの制御を許可する
};

include "/etc/rndc.key"; ―――――――――――――――――――――――― 認証鍵を外部ファイルから取り込む
```

リスト1● サーバ側のBIND設定ファイル「/etc/named.conf」の設定例

```
key "rndc-key" {
        algorithm       hmac-md5;
        secret "UDnMeMKGUhAKtbtQJuKXNReWhGqar0GeQplcpxyHLOBqsMIWicLoRSxPsDvn";
};
```

リスト2● クライアント側の認証鍵設定ファイル「/etc/rndc.key」の設定例

```
options { ―――――――――――――――――――――――――――――― デフォルト設定
        default-server  localhost;
        default-key     "localhost-key";
};

server localhost { ―――――――――――――――――――――――――― ローカルホストを制御するための設定
        key     "localhost-key";
};

server 192.168.0.10 { ―――――――――――――――――――――――― リモートネームサーバを制御するための設定
        key     "example-key";
};

key "localhost-key" { ―――――――――――――――――――――――― ローカルホスト用の認証鍵の設定
        algorithm       hmac-md5;
        secret "LZNOm6odk3FYAsBx1XwO8PB0PjIGeX9Bp9pAOZJNEVMTJLWfZIaIOq59BtGd";
};

key "example-key" { ―――――――――――――――――――――――― リモートネームサーバ用の認証鍵の設定
        algorithm       hmac-md5;
        secret "UDnMeMKGUhAKtbtQJuKXNReWhGqar0GeQplcpxyHLOBqsMIWicLoRSxPsDvn";
};
```

リスト3● 「/etc/rndc.conf」ファイルの設定例

を見ることができないように、アクセス権を設定しておく。namedをユーザー「named」で動作させている場合は次のように設定すればよい。

```
# chown named /etc/rndc.key
  └ 所有者を「named」に変更する
```

```
# chmod 600 /etc/rndc.key
  └ 所有者のみがファイルにアクセスできるようにする
```

この「/etc/rndc.key」ファイルは、「rndc-confgen」プログラムを使用して自動的に作成することもできる。例えば、鍵のサイズが256ビット、作成するファイルの所

有者をnamedに設定した「/etc/rndc.key」ファイルを自動的に作成するには、次のようにコマンドを発行する。

```
# /usr/sbin/rndc-confgen -a -b 256 -u named
```

クライアント側の設定は/etc/rndc.confファイルで行う

次に、クライアント側の設定を行う。クライアント側では、前ページのリスト3のように「/etc/rndc.conf」ファイルで、リモートネームサーバのアドレスと、サーバ側で設定した認証鍵を設定する。

そして、このファイルにrootのみがアクセスできるように、次のようにして所有者とアクセス権の設定をしておく。

```
# chown root /etc/rndc.conf
  └ 所有者を「root」に変更する
# chmod 600 /etc/rndc.conf
  └ 所有者のみがファイルにアクセスできるようにする
```

rndcを利用してみる

これで、rndcの設定は完了である。サーバ側でBIND 9を再起動させ、クライアント側でrndcコマンドを発行させてみよう。「-s」オプションで、制御を行うネームサーバのホストを指定し、その後にrndcのオプションを指定する。

```
# /usr/sbin/rndc -s 192.168.0.10 status
  └ 192.168.0.10上のネームサーバのステータスを表示
number of zones: 4
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
server is up and running
```

rndcコマンドで使用できるオプションには、表1のものがある。

オプション	説明
reload	設定ファイルとゾーンデータファイルをリロードする
reload <zone>	指定したゾーンのみをリロードする
refresh <zone>	指定したゾーンに対して、直ちにゾーン転送を開始する
reconfig	設定ファイルと変更したゾーンデータファイルをリロードする
stats	「named.stats」ファイルに統計情報を書き込む
querylog	クエリごとのログをsyslogに出力するかどうかを切り替える
dumpdb	「named_dump.db」ファイルにキャッシュデータをダンプする
stop	ダイナミックアップデート情報を書き込んでからサーバを停止する
halt	ダイナミックアップデート情報を書き込まずにサーバを停止する
flush	サーバのキャッシュデータをフラッシュする
status	サーバのステータスを表示する

表1● rndcコマンドで指定できるオプションの一部

Q12 ファイアウォールでネームサーバを保護するには？

ネームサーバを不正アクセスから保護するため、ファイアウォールを適切に設定してネームサーバへのアクセスを制限したい。

A DNSの問い合わせ以外で発生するパケットをファイアウォールでフィルタリングする

ファイアウォールでネームサーバを保護するためには、外部と通信する必要のあるネームサーバへのトラフィックを、ファイアウォールで適切にフィルタリングする必要がある。

ここでは、ファイアウォールで保護すべきネームサーバとして、図2のように、外部ネームサーバへの問い合わせを行うためのキャッシングネームサーバ（ローカルネームサーバ）と、自ゾーンの情報を外部に公開するための権威ネームサーバの2つが存在すると仮定する。

キャッシングネームサーバをファイアウォールで保護する

まず、図2のキャッシングネームサーバによって発生するトラフィックを考えてみる。キャッシングネームサ

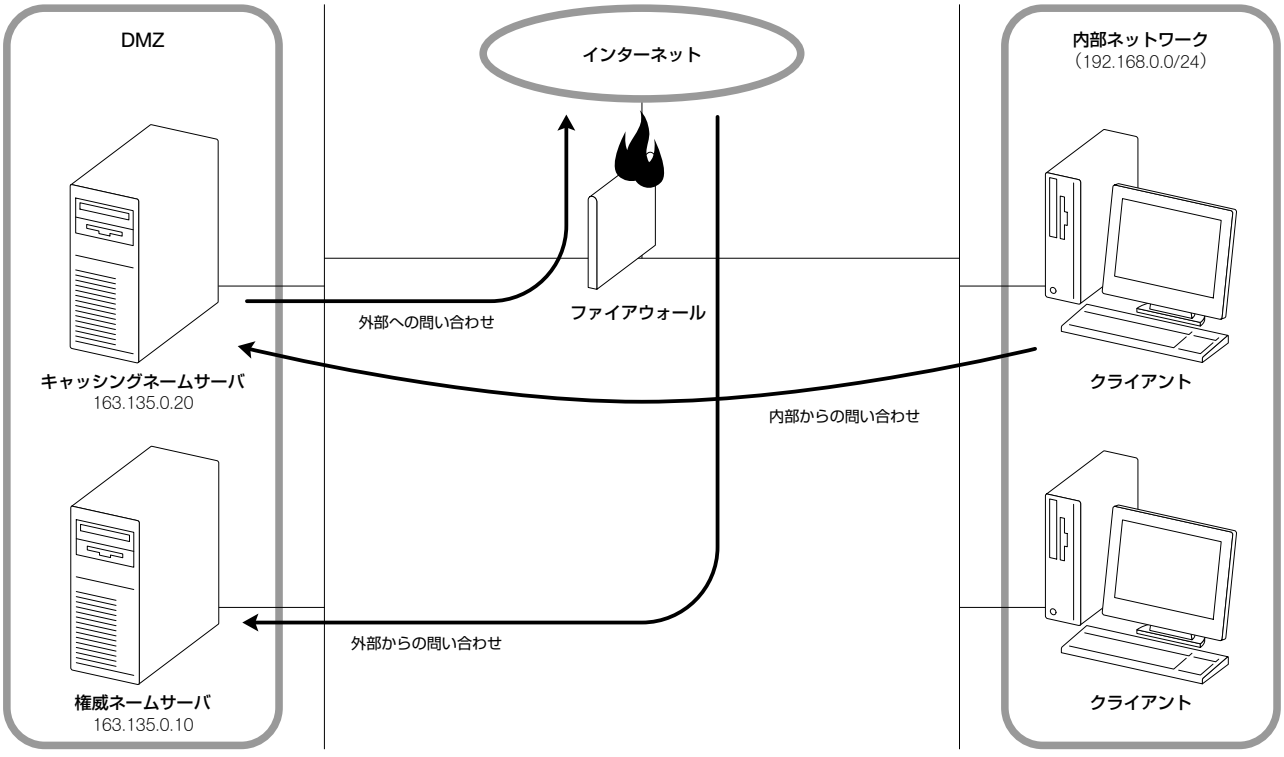

図2● ファイアウォールでキャッシングネームサーバと権威ネームサーバを保護する

ーバからは、外部のさまざまなネームサーバに対して問い合わせが発生するが、外部から問い合わせを受けることはない。また、ゾーン転送などの問い合わせが発生することもない。そこで、ファイアウォールでは、表2のように、キャッシングネームサーバから外部のネームサーバの53番ポートに対するパケットとその返答パケットのみを通過するように設定する。ここで、キャッシングネームサーバが比較的最近のBINDを使用しているのであれば、問い合わせの送信元ポートとして非特権ポートが使用されるので、ファイアウォールでは、キャッシングネームサーバの1024以上のポートからのパケットを許可するように設定する。しかし、古いBINDを使用している場合には、問い合わせの送信元ポートとして53番ポートを使用するため、53番ポートからのパケットを許可するように設定しなければならない。もちろん、その前に新しいBINDを使用することをお勧めする。また、通常の問い合わせはUDPを使用するが、回答データのサイズが大きくなる場合は、TCPで接続する必要があるため、UDPだけでなく、TCPも通すように設定しておく。

そして、内部ネットワーク上のクライアントからキャッシングネームサーバに対して問い合わせが行えるように、表3のように、内部ネットワークで使用しているアドレスからキャッシングネームサーバの53番ポートあてのパケットの通過を許可するように設定する。

プロトコル	送信元アドレス	送信元ポート	宛先アドレス	宛先ポート
UDP	163.135.0.20	1024以上	ANY	53
UDP	ANY	53	163.135.0.20	1024以上
TCP	163.135.0.20	1024以上	ANY	53
TCP	ANY	53	163.135.0.20	1024以上

表2● キャッシングネームサーバから外部ネームサーバへの問い合わせを許可するフィルタリングの設定例

プロトコル	送信元アドレス	送信元ポート	宛先アドレス	宛先ポート
UDP	192.168.0.0/24	1024以上	163.135.0.20	53
UDP	163.135.0.20	53	192.168.0.0/24	1024以上
TCP	192.168.0.0/24	1024以上	163.135.0.20	53
TCP	163.135.0.20	53	192.168.0.0/24	1024以上

表3● 内部ネットワークからキャッシングネームサーバへの問い合わせを許可するフィルタリングの設定例

権威ネームサーバをファイアウォールで保護する

次に、図2の権威ネームサーバに対して発生するトラフィックを考えてみる。外部からの問い合わせでは、権威ネームサーバの53番ポートあてのUDPトラフィックが

発生するので、このパケットを通すように設定する。最近のBINDであれば、問い合わせパケットの送信元ポートには、1024以上のポートが使用されるが、問い合わせ元が古いBINDを使用している場合も考えられるので、53番ポートからの問い合わせも通すように設定しておく（表4）。

ただし、自ゾーンへの問い合わせに対する回答メッセージのサイズが512バイトより大きくなる場合は、TCPを使用した問い合わせが発生する場合があるので、このような場合には、表5のように、TCPでの問い合わせも許可するように設定する。

また、外部のネームサーバに、自身が管理するゾーンのセカンダリをお願いしている場合には、ゾーン転送用のトラフィックも許可するように設定しなければならない。ゾーン転送で発生するトラフィックには、ゾーンデータが更新されたことを知らせるNOTIFYメッセージ、ゾーンが更新されたかどうかを確認するためのSOAレコードの問い合わせ、そして、ゾーン転送があり、表6のトラフィックを許可するように設定する。

BINDが送出するUDPの送信元ポートを固定する

BINDでは、問い合わせに使用するUDPの送信元ポートやアドレスを指定することもできる。リスト4のように、「query-source」サブステートメントを使用すると、ネームサーバが複数のIPアドレスを持っていた場合に、送信元IPアドレスとして使用するアドレスを指定したり、通常は1024以上の値で自動的に決められる送信元ポートを非特権ポートの範囲内で固定させたりすることができる。同様に、ゾーン転送の際のSOAレコードの問い合わせや差分ゾーン転送で使用するアドレスとポートは「transfer-source」サブステートメントで、NOTIFYメッセージの送信で使用するアドレスとポートは「notify-source」サブステートメントで指定できる。これらの機能を使用すれば、ファイアウォールでの設定をさらにシンプルにすることが可能だ。ただし、ポートの指定はUDPのみに適用され、TCPの送信元ポートは固定することができないので注意してほしい。

NTTデータ　馬場達也

プロトコル	送信元アドレス	送信元ポート	宛先アドレス	宛先ポート
UDP	ANY	1024以上	163.135.0.10	53
UDP	163.135.0.10	53	ANY	1024以上
UDP	ANY	53	163.135.0.10	53
UDP	163.135.0.10	53	ANY	53

表4●　外部から権威ネームサーバへの問い合わせを許可するフィルタリングの設定例

プロトコル	送信元アドレス	送信元ポート	宛先アドレス	宛先ポート
UDP	ANY	1024以上	163.135.0.10	53
UDP	163.135.0.10	53	ANY	1024以上
UDP	ANY	53	163.135.0.10	53
UDP	163.135.0.10	53	ANY	53
TCP	ANY	1024以上	163.135.0.10	53
TCP	163.135.0.10	53	ANY	1024以上

表5●　外部から権威ネームサーバへの問い合わせを許可するフィルタリングの設定例（回答のサイズが大きくなる場合）

プロトコル	送信元アドレス	送信元ポート	宛先アドレス	宛先ポート
NOTIFYメッセージ				
UDP	（プライマリ）	1024以上	（セカンダリ）	53
UDP	（セカンダリ）	53	（プライマリ）	1024以上
SOAレコードの問い合わせ				
UDP	（セカンダリ）	1024以上	（プライマリ）	53
UDP	（プライマリ）	53	（セカンダリ）	1024以上
UDP	（セカンダリ）	53	（プライマリ）	53
UDP	（プライマリ）	53	（セカンダリ）	53
ゾーン転送				
TCP	（セカンダリ）	1024以上	（プライマリ）	53
TCP	（プライマリ）	53	（セカンダリ）	1024以上

表6●　セカンダリにゾーン転送を行う場合のフィルタリングの設定例

```
options {
        directory "/var/named/";                          問い合わせの送信元ポートを1053に固定
        query-source address 163.135.0.10 port 1053;
        transfer-source 163.135.0.10 port 1053;           ゾーン転送の際のSOAレコードの問い合わせや差分ゾーン転送で使用する送信元ポートを1053に固定
        notify-source 163.135.0.10 port 1053;             NOTIFYメッセージの送信元ポートを1053に固定
};
```

リスト4●　UDPの送信元ポートを固定するための設定例