三重大学教育学部研究紀要　第58巻　自然科学　(2007)　13－20頁

1台のマシン上でのコンテンツDNSサーバとキャッシュDNSサーバの設定方法

山守一徳*

Setting up Content DNS-Server and Cache DNS-Server on One Machine

Kazunori YAMAMORI

要　旨

DNSサーバに対するDOS攻撃（サービス不能攻撃）が報告されるようになり、従来の設定のままでDNSサーバを運営していると、被害を受ける恐れがある。その対策として、DNSサーバをコンテンツサーバとキャッシュサーバに分離することが提唱されている。しかし、DNSサーバとして有名な“BIND”ソフトは、1台のマシン上で分離設定するには工夫を必要とする。教育学部で運営しているDNSサーバ（minerva）も例外ではない。この度、minervaを更新するに当たり、DNSサーバの設定も改善させ、“BIND9”を用いて、1台のマシン上でコンテンツサーバとキャッシュサーバに分離することを行った。その設定方法について報告する。

1．はじめに

DNSサーバは、IPアドレスとホスト名を結び付けるサーバであり、ネットワーク管理者がサーバ内の設定を行い、一般利用者は、ネットワーク設定時に各自の端末の中にDNSサーバのIPアドレスを設定するのみで、普段使用している状況を感じさせないサーバである。一般利用者が最も恩恵を受けるのは、Internet Explorer等のWebブラウザを使う時に、URLを入力する欄において、http://www.xxx.yyy.co.jp/hello.html等と入力すると、そのサーバのWebページが表示されてくる時であろう。DNSサーバの御陰で、www.xxx.yyy.co.jpのマシンのIPアドレスがわかり、http://の後ろに直接IPアドレスを打たなくても良いからである。逆に、DNSサーバが止まっている時には、www.xxx.yyy.co.jpのマシンのIPアドレスが、123.100.200.30であったとすると、http://123.100.200.30/hello.htmlと打てば、そのサーバのWebページが表示されてくる。「Webページが表示されないのですが、、、」という問い合わせを受ける時があるが、その時DNSサーバが止まっていることが原因でIPアドレスが検索できずにWebページが表示されないという場合もある。http://123.100.200.30/hello.htmlと打てば表示されて、http://www.xxx.yyy.co.jp/hello.htmlと打てば表示されない場合は、原因はDNSサーバを疑うと良い。

このDNSサーバに対してサービス不能に陥れる攻撃（DOS攻撃）をハッカー等が仕掛けてくることがある。DOS攻撃を受けるとwww.xxx.yyy.co.jpのようなFQDN（Full Quality Domain Name）からIPアドレスを答えてくれなくなることが起きる。これはそのDNSサーバを利用している人々へ迷惑を掛けてハッカー等が楽しんでいるだけでなく、管理者にとって怖いのは、サービス不能になった

*三重大学教育学部情報教育課程

DNSが管理していたデータとは違う偽のデータをインターネット上へ蔓延させられ、別のIPアドレスの方へ第3者のユーザを導くことをされることである。例えば、＊.edu.mie-u.ac.jpのホスト名に対して偽のIPアドレスがインターネット上に蔓延されると、www.edu.mie-u.ac.jpのWebページを見ようとしている第3者のユーザは、偽のIPアドレス上のWebページを本物と思い込んで読んでしまうことが起きる。ハッカー達は、偽のWebページへ導いて、そこでログインパスワードを打ち込ませてパスワードを盗んだり、クレジット番号や誕生日等を打ち込ませて、個人情報を盗んだりする。

そこで、DNSサーバを堅牢に運営する必要があり、その設定方法を報告する。

2．コンテンツDNSサーバとキャッシュDNSサーバ

DNSサーバの中に機能には、コンテンツサーバとキャッシュサーバ（キャッシュサーバはリゾルバサーバとも呼ばれる）の2つの機能がある。コンテンツサーバとキャッシュサーバを分離せず、1つのプロセスで稼動させている場合には、DOS攻撃を受けてコンテンツサーバ機能がStopすると、キャッシュサーバ機能もStopしてしまう。逆に、キャッシュサーバ機能に対しDOS攻撃を受けた場合には、コンテンツサーバ機能もStopしてしまう。よって、コンテンツサーバとキャッシュサーバを分離した方が堅牢となる。

(1) コンテンツDNSサーバ

そのDNSサーバの管理下であるドメイン内のマシンに関して、ホスト名とIPアドレスの組み合わせの情報を保持管理しているのが、コンテンツDNSサーバである。minervaの場合には、edu.mie-u.ac.jpのドメイン内のマシンを管理している。この情報をゾーン情報と呼ぶ。ゾーン情報は、別のDNSサーバへ一括で転送し（ゾーン転送と呼ぶ）、情報を共有し合ってどちらのサーバでも答えられる状況を作り出している。共有し合う時に、主となるサーバはプライマリサーバ、従となるサーバはセカンダリサーバと呼ばれる。

このゾーン転送は負荷の掛かる処理であり、DOS攻撃の一つに、ゾーン転送の要求を相手のDNSサーバへ多数連続して出すという攻撃方法がある。短時間の間に集中して頻繁に要求を出されるとサーバは応答する処理が追いつかなくなり、利用者から見てサーバが応答を返さずにダウンしてしまったように見える。

本来、ゾーン情報を共有して保持していて欲しい相手は、自分から身近な特定のDNSサーバであるので、ゾーン転送の要求に答えるのは、そのサーバからだけにして良い。minervaの場合には、mie-u.ac.jpのドメインを管理している総合情報処理センターのDNSサーバ（dns.cc.mie-u.ac.jp）にセカンダリサーバを担当していただいている。そのセカンダリサーバに対してゾーン転送を許可すれば良い。

(2) キャッシュDNSサーバ

ホスト名からIPアドレスを返すようにクライアント端末から要求された時に、それまでに調べた情報をキャッシュに保存しているので、そのキャッシュの中から見つけて返すか、または、ルートのDNSサーバから始まってより下部のDNSサーバへ再帰的に調べてIPアドレスを返すサーバである。IPアドレスからホスト名を探す逆引きと呼ばれる探索も行う。

例えば、www.xxx.yyy.co.jpのマシンのIPアドレス再帰的に探索する場合の流れは以下のようになる。

Step1：ルートドメインを担当しているDNSサーバへ.jpドメインを管理しているDNSサーバのIPアドレスを尋ねる。

Step2：Step1からIPアドレスを知って、jpドメインを担当しているDNSサーバへco.jpドメインを管理しているDNSサーバのIPアドレスを尋ねる。

Step3：Step2からIPアドレスを知って、co.jpドメインを担当しているDNSサーバへyyy.co.jpドメインを管理しているDNSサーバのIPアドレスを尋ねる。

Step4：Step3からIPアドレスを知って、yyy.co.jpドメインを担当しているDNSサーバへxxx.yyy.co.jpドメインを管理しているDNSサーバのIPアドレスを尋ねる。

Step5：Step4からIPアドレスを知って、xxx.yyy.co.jpドメインを担当しているDNSサーバへwww.xxx.yyy.co.jpのIPアドレスを尋ねる。

以上で、www.xxx.yyy.co.jpのIPアドレスを知ることができる。一度知った情報は、キャッシュの中へ保存され、それ以後の同じ問い合わせに対しては即返答が返るようになる。

このキャッシュDNSサーバの再帰的に探索する処理がサーバに取って負担が大きい。身近なユーザから依頼された分だけならば余裕の仕事量であるが、インターネット上の第3者のユーザからも多数探索依頼を受けると過負荷状態に陥ってしまう。そうなると本来の身近なユーザに対して返答を返すことができなくなってしまう。そこで、このキャッシュDNSサーバへの問い合わせも特定のIPアドレスの範囲からだけ受け付けるようにする。minervaの場合には、大学内のIPアドレスからのみの探索要求を受け付けるようにする。

3．BIND

BINDはDNSサーバとして採用される最も有名なソフトである。BINDは、通常、コンテンツサーバとキャッシュサーバの両方を1つのプロセスで動かしている。設定の仕方によっては、コンテンツサーバのみ、またはキャッシュサーバのみの機能になるが、コンテンツサーバとキャッシュサーバをそれぞれ別プロセスにして1台のマシン上で稼動させるには工夫を要する。キャッシュサーバ用にdnscache、コンテンツサーバ用にtinydnsとプログラムが分かれた構造になっているdjbdnsというソフトが存在するが、tinydnsはBINDのセカンダリサーバからのゾーン転送要求に正しく応答しないとも言われており、そのためaxfrdnsというプログラムがさらに必要と言われ、普及度が低い。コンテンツサーバ用には、NSDというソフトも存在するが、こちらも普及度が低い。そこで、BINDのままで別ソフトへ移行せずに継続使用することを考える。最も堅牢になるのが、コンテンツサーバのプロセスとキャッシュサーバのプロセスを別々のマシンで稼動させることである。別々のマシンで稼動させておけば、DOS攻撃を受けた場合に片側のみサービス不能になったとしてももう片側は稼動できるからである。しかし、マシンが2台必要となるため、1台で済ませたいのが実状である。そこで、1台のマシン上で、BIND9を用いてコンテンツサーバとキャッシュサーバをそれぞれ別プロセスで稼動させる設定方法を紹介する。1台のマシン上で2つのプロセスを稼動させるには、IPアドレスを2つ用意し、各プロセスは個々に異なるIPアドレスから要求を受け付けるようにする。2つのIPアドレスが必要になってしまうのは、コンテンツサーバとキャッシュサーバはどちらも53番という同じポート番号を使うからである。IPアドレスを2つ用意するには、ネットワークケーブルの物理的な差し込み口を2つ用意するか、または、1つの口で、2つのIPアドレスを設定する。IPアドレス自身の設定方法はここでは省略する。

(1) コンテンツ DNS サーバ

named のプロセスを起動させる時に、/usr/sbin/named -u named -c /etc/named.conf start のように -c で設定ファイルを指定して起動させる。設定ファイル named.conf の中味は図 1 のようになる。

```
options {
directory "/var/named";    // ゾーンファイルと DNS 探索 root ファイルの置き場所
dump-file "/var/named/data/contents-cache-dump. db";
statics-file "/var/named/data/contents-named-stats. txt";
recursion no;
listen-on { コンテンツ DNS サーバの IP アドレス; };
version "Contents Server";    // セキュリティ上、バージョンは外部に見せない
allow-transfer { none; };
};
controls {
inet コンテンツ DNS サーバの IP アドレス allow {localhost; } keys { rndckey; };
}
zone "." IN {
type hint;
file "/dev/null";  // root サーバは不要
};

//  後は管理しているゾーン分が続く。例えば、
zone "edu. mie-u. ac. jp"  {
type master;
file "edu. mie-u. ac. jp を管理するゾーンファイル名" ;
allow-transfer { 133. 67. 0. 0/16; };
allow-query { any; };
};
zone "95. 67. 133. in-addr. arpa" {
type master;
file "133. 67. 95 を管理するゾーンファイル名";
allow-transfer { 133. 67. 0. 0/16; };
allow-query { any; };
};
//  この下には 133. 67. 95 以外の管理している部分が繰り返される。
//  繰り返し部分

zone "mie-u. ac. jp"  {
type slave;
file "mie-u. ac. jp を管理するゾーンファイル名" ;
masters { 133. 67. 1. 1; };
allow-query { any; };
};
zone "67. 133. in-addr. arpa"  {
type slave;
file "133. 67 を管理するゾーンファイル名" ;
masters { 133. 67. 1. 1; };
allow-query { any; };
};

//  最後に
include "/etc/rndc. key";
```

図 1　コンテンツ DNS サーバ用の設定ファイルの例

設定のポイントは、“. ”に対するゾーンファイル名が存在しないことである。

(2) キャッシュDNSサーバ

named のプロセスを起動させる時に、/usr/sbin/named -u named -c /etc/cache.conf start のように -c で設定ファイルを指定して起動させる。設定ファイル cache.conf の中味は図2のようになる。

```
cl my-network {
133. 67. 0. 0/16;
};

options {
directory “/var/named”;
dump-file “/var/named/data/cache-dump. db”;
statistics-file “/var/named/data/named-stats. txt”;
pid-file “/var/run/named/cache-named. pid”;
recursion yes;
listen-on { キャッシュ DNS サーバ用の IP アドレス; };
version “Cache Server”; // セキュリティ上、バージョンは外部に見せない
allow-query {
  localhost;
   my-network;
};
};
controls{
inet キャッシュ DNS サーバ用の IP アドレス allow { localhost; } keys { rndckey; };
}
zone “. ” IN {
  type hint;
   file “named. ca”;
};
zone “localhost” IN {
  type master;
   file “localhost. zone”;
  allow-update { none; };
};
zone “0. 0. 127. in-addr. arpa” IN {
  type master;
   file “named. local”;
  allow-update { none; };
};
zone “255. in-addr. arpa” IN {
  type master;
   file “named. broadcast”;
  allow-update { none; };
};
zone “0. in-addr. arpa” IN {
  type master;
   file “named. zero”;
  allow-update { none; };
};
zone “10. in-addr. arpa” IN {
```

```
  type master;
   file "private_dummy. rev";
  allow-update { none; };
};
zone "16. 172. in-addr. arpa" IN {
  type master;
   file "private_dummy. rev";
  allow-update { none; };
};

//  あとは 31. 172. in-addr. arpa まで繰り返し
//  繰り返し部分

zone "168. 192. in-addr. arpa" IN {
  type master;
   file "private_dummy. rev";
  allow-update { none; };
};

zone "254. 169. in-addr. arpa" IN {
type master;
file "private_dummy. rev";
allow-update { none; };
};
include "/etc/rndc. key";
```

図2　キャッシュ DNS サーバ用の設定ファイルの例

ゾーンファイルまたはマスタファイルと呼ばれる設定ファイルの中身の書き方は、コンテンツサーバとキャッシュサーバに分離しているか否かに依存しないため、ここでは説明を省略する。

4．自動起動方法

Linux（CentOS4. 2）でパッケージインストールすると、/etc/init.d/named のファイルが、起動用のスクリプトファイルとなっている。インストールに必要なパッケージは、bind、bind-utils、bind-libs、caching-nameserver である。この起動用のスクリプトファイルの中は、start させる時には、/usr/sbin/named -u named のコマンドで起動させ、stop させる時には、/usr/sbin/rndc stop のコマンドで停止させ、再起動させる時には、/usr/sbin/rndc reload のコマンドで再起動するように書かれている。この記述のままでは、1 つの named プロセスをコントロールできるだけであるので、今回のように 2 つの named プロセスを稼動させたい場合には、このスクリプトファイルも 2 つ用意する必要がある。

作成したスクリプトファイルは、/etc/init.d/に置く。なお、パッケージインストールすると、named のユーザが作られる。SELinux が有効になっている時には、/usr/sbin/named を named のユーザで起動する必要がある。Root ユーザで起動したりすると syslog の中にエラーが出力されてしまう。

(1) コンテンツ DNS サーバ用の起動スクリプトファイル

デフォルトで用意されるスクリプトファイルの中の、/usr/sbin/rndc stop と記述されている箇所を /usr/sbin/rndc -s コンテンツ DNS サーバ用の IP アドレス stop と変更し、/usr/sbin/rndc reload と記

述されている箇所を/usr/sbin/rndc -s コンテンツDNSサーバ用のIPアドレス reloadと変更する。

(2) キャッシュDNSサーバ用の起動スクリプトファイル

デフォルトで用意されるスクリプトファイルのままでは、コンテンツDNSサーバのプロセスを制御する操作と重なってしまうために、以下の箇所を変更する。

(イ) prog＝"named" と記述されている箇所をprog＝"named_cache" と変更する。

(ロ) /sbin/pidof named と記述されている箇所を/sbin/pidof named_cache と変更する。

(ハ) 設定ファイルがnamed.confでなく、キャッシュDNSサーバ用にはcache.confという名前を用いた場合、${ROOTDIR}/etc/named.conf と記述されている箇所を${ROOTDIR}/etc/cache.conf と変更する。

(ニ) daemon /usr/sbin/named -u named ${OPTIONS};と記述されている箇所をdaemon /usr/sbin/named -u named -c /etc/cache.conf ${OPTIONS};と変更する。

(ホ) /var/lock/subsys/named と記述されている箇所を/var/lock/subsys/named_cache と変更する。

(ヘ) /usr/sbin/rndc stop と記述されている箇所を/usr/sbin/rndc -s キャッシュDNSサーバ用のIPアドレス stop と変更する。

(ト) /usr/sbin/rndc reload と記述されている箇所を/usr/sbin/rndc -s キャッシュDNSサーバ用のIPアドレス reload と変更する。

5. 考 察

以上のように設定して稼動させた場合の問題点としては、BINDをアップデートする時に、Yumなどを使ってパッケージアップデートを使うと、BINDの起動用スクリプトファイルが上書きされてしまうという問題がある。BINDのアップデートが頻繁に起きる場合には、上書きされた場合の修復作業が手間であるので、自動でアップデートされないようにするためには、/etc/yum.confのファイルの中に、exclude＝bind＊という行を追加して、bind＊のパッケージのみアップデートされないようにする方法がある。しかし、アップデートを長期間しない訳にもいかず、今後はBINDの利用を諦めてdjbdnsとNSDを使うように変更していくことが考えられる。

キャッシュDNSサーバ用のIPアドレスとコンテンツDNSサーバ用のIPアドレスをどう割り当てるかについて、これまでのDNSサーバに割り当てていたIPアドレスをコンテンツDNSサーバ用に用いた。これは、学外からアクセスされるのは、コンテンツDNSサーバの機能だからである。総合情報処理センターに存在しているファイアウォールでは、コンテンツDNSサーバのIPアドレスに対しは、通過可能な設定になっており、キャッシュDNSサーバ用のIPアドレスに対しては、遮断される設定になっている。キャッシュDNSサーバの機能は、学内のみからアクセスされ、学外からは攻撃を避けるため遮断されていれば良いからである。

キャッシュDNSサーバ用のIPアドレスを新設すると、学部内の端末の中で、DNSサーバのIPアドレスを指定していた欄を書き直す必要がある。書き直さずにそのままにしている端末は、コンテンツDNSサーバのIPアドレスを指定していることになるので、コンテンツDNSサーバへ問い合わせを行うことになり、学部で管理しているIPアドレスの範囲のみ、FQDNからIPアドレスを引くことができ、それ以外のFQDNについて問い合わせてもIPアドレスを引くことはできなくなるので注意が必要である。

6．まとめ

BIND を用いて、1 台のマシン上でコンテンツ DNS サーバとキャッシュ DNS サーバをそれぞれ別プロセスで稼動させる設定方法を紹介した。IP アドレスを 2 つ用意し、各プロセスは個々に異なる IP アドレスから要求を受け付けるようにした方法である。キャッシュ DNS サーバは学内のみアクセス可とし、コンテンツ DNS サーバは、学外からもアクセス可とするが、ゾーン転送については特定のセカンダリサーバからのみアクセス可とする。

Phishing と呼ばれる第 3 者を騙して本物の Web サイトであると思わせる詐称サイトがインターネット上に現れたりしており、DNS サーバを正しく稼動させることがより重要なことになってきている。DNS サーバが正しく稼動していればそのような詐称サイトは少なくなると思われ、サーバを堅牢にして運営したいものである。