Впервые опубликовано на developerWorks 01.12.2005

Экзамен LPI: Domain Name System (DNS, Доменная система имен)

Администрирование Linux, средний уровень (LPIC-2) тема 207

DNS

Доменная система имен (Domain Name System) позволяет достаточно удобным образом обращаться к серверам по имени, а не IP-адресу, всем пользователям TCP/IP приложений. Berkeley Internet Name Domain (BIND) представляет из себя сервер named, отвечающий на запросы о связи IP-адреса с символическим именем (или наоборот, а также другую информацию). Со стороны клиента системы DNS имеется набор библиотек resolver, позволяющих приложениям связываться с DNS серверами. В пакет BIND также входит ряд клиентских приложений для конфигурирования, осуществления запросов и отладки BIND 9 сервера: `dig`, `nslookup`, `host`, and `rndc` (ранее `ndc`). В сущности, эти приложения вызывают те же самые библиотеки, что и другие клиентские приложения, непосредственно реагируя на запросы DNS серверов.

BIND

На время написания данного руководства текущей версией BIND была 9.3.1. Первая стабильная версия серии BIND 9 была выпущена в октябре 2000. На некоторых старых инсталляциях вы можете найти BIND 8, который продолжает поддерживаться путем выпуска security patches (текущая версия 8.4.6), но, как правило, обновляются они до BIND 9 везде, где это возможно. Самые архаичные системы могут иметь BIND 4, но их рекомендуется обновлять как можно скорее, поскольку поддержка BIND 4 прекращена.

Все версии BIND могут быть получены от Internet Systems Consortium (ISC; см. ссылку в разделе Ресурсы). Документацию и другие ресурсы, связанные с BIND, также можно найти на этом сайте.

Поскольку основным для данной темы является знание конфигурации BIND 8 и мы далее сосредотачиваемся именно на BIND 8, мы рекомендуем ознакомиться с информацией по BIND 8 на сайте ISC, перед сдачей экзамена LPI 202.

Другие ресурсы

Как и для большинства Linux-приложений, крайне полезным является изучение страниц руководств man по всем обсуждаемым утилитам. Версии и опции могут различаться от версии к версии утилиты или ядра, или между различными дистрибутивами Linux. Более глубокую информацию можно найти в большом количестве очень полезных HOWTO из Linux Documentation Project (см. ссылку в разделе Ресурсы). Был опубликован ряд полезных книг, в особенности хотелось бы

отметить TCP/IP Network Administration издательства O'Reilly (постарайтесь найти последнее издание; см. ссылку в разделе Ресурсы).

Для получения информации именно о DNS и BIND очень хорошим источником может являться DNS and BIND, Fourth Edition издательства O'Reilly (см. ссылку в разделе Ресурсы); на ее 622 страницах можно найти более детальную информацию, чем в данном руководстве. Другие издательства также выпустили ряд изданий, посвященных BIND.

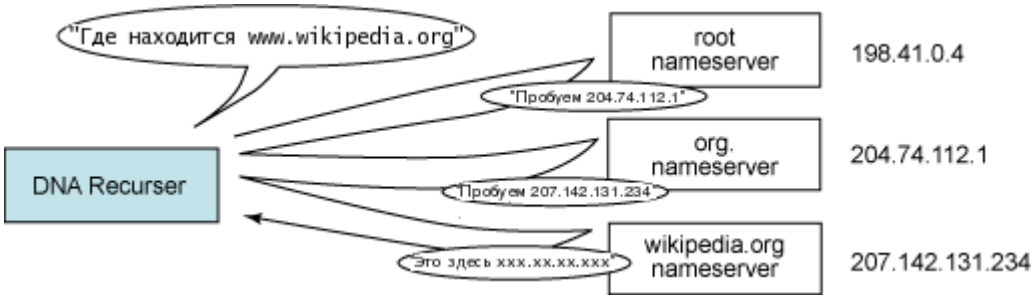
Запросы доменной системы имен

Топология DNS

DNS представляет из себя иерархическую систему доменных зон. Каждая зона предоставляет ограниченный набор отображений в доменные имена, входящие в ее поддомен. Запрошенный сервер пошлет запрос более высокому в иерархии серверу в случае, если он не в состоянии найти требуемое отображение и, в случае необходимости, будет следовать указаниям о перенаправлении, пока не найдет правильного соответствия (или не определит, что данный запрос не может быть разрешен, выдав ошибку). Когда локальный сервер named получает ответ на посланный DNS запрос, он кэширует его на период времени, указанный в конфигурации (это скорее часы, чем секунды или дни). Вследствие кэширования DNS запросов совокупная сетевая загрузка значительно снижается, особенно для серверов верхнего уровня домена (top-level-domain -- TLD). Статья о DNS в Wikipedia (см. ссылку в разделе Ресурсы) -- отличная стартовая точка для понимания архитектуры в целом. В этом руководстве мы приводим диаграмму из этого источника, предоставленную в свободный доступ (см. рис. 1 далее).

Диаграмма прохождения гипотетического DNS запроса позволит взглянуть на процесс его обслуживания в целом. Положим, ваша локальная машина хочет получить доступ к машине с доменным именем www.wikipedia.org. Для поиска соответствующего IP адреса ваша машина должна обратиться к локальному серверу имен, указанному в конфигурации клиентской машины. Этот сервер может работать на той же самой машине, может располагаться на DNS сервере вашей локальной сети (LAN) или же предоставляться вашим интернет-провайдером (ISP). Во всех этих случаях это будет какой-то экземпляр BIND named. Этот локальный сервер имен в первую очередь проверит свой кэш и в случае, если там такой информации нет, выполнит следующие шаги, отображенные на диаграмме:

Рисунок 1. Пример рекурсии DNS



Следует понимать, что "DNS Recurser" -- это текущий DNS сервер (named), а не клиентское приложение, которое с ним общается.

DNS использует TCP и UDP на порту 53 для обслуживания запросов. Практически все DNS запросы представляют из себя одиночные UDP запросы от клиента, порождающие одиночные UDP ответы от сервера.

Откуда приложение знает, где найти DNS сервер

Конфигурирование клиентского приложения для доступа к DNS серверу (серверам) достаточно просто. Вся конфигурация содержится в файле /etc/resolve.conf, в котором указаны один или более "локальных" DNS серверов. Вы можете вручную сконфигурировать /etc/resolve.conf на подключение к известным вам DNS серверам; однако, если вы используете DHCP для конфигурирования клиента, в ходе DHCP handshaking'a эта информация добавляется в /etc/resolve.conf автоматически (вы по-прежнему можете читать его и даже модифицировать установки, сделанные DHCP, но они будут вновь восстановлены после перезагрузки). Код библиотеки с установками, сделанными /etc/resolv.conf, называется "DNS resolver."

Если в /etc/resolv.conf указано более одного DNS сервера, вторичный и третичный DNS сервера будут использоваться в случае, если ответ от первичного сервера не поступает в течение указанного периода времени. Максимальное количество серверов, которое можно указать в конфигурации -- три.

Прежде всего, файл /etc/resolv.conf содержит записи следующего вида `nameserver <IP-addr>`. Некоторые другие записи могут изменять ответы на посланные запросы. Например, директивы `domain` и `search` расширяют имена, не содержащие точек (машины в локальной сети). Директива `options` позволяет вам изменять время ожидания ответа от DNS сервера, включать режим отладки при расширении до полного доменного имени и изменять другие аспекты работы DNS resolver'a. Например, на одной из моих машин конфигурация такова:

Листинг 1. Модификация опций в файле конфигурации доступа к DNS серверам

```
# cat /etc/resolv.conf
search gnosis.lan
nameserver 0.0.0.0
nameserver 192.168.2.1
nameserver 151.203.0.84
options timeout:3
```

Первая директива указывает, что машины в локальной сети входят во внутренний домен gnosis.lan, таким образом, короткое имя bacchus может быть расширено в bacchus.gnosis.lan. Несколько доменов, разделенных пробелами могут быть перечислены в директиве `search`.

Затем я перечислил несколько DNS серверов. Первый из них -- локальная машина, на которую можно ссылаться как на 0.0.0.0 или через официальный IP адрес, но не loopback-адрес. Следующая директива, `nameserver`, указывает на мой домашний роутер, соединяющий мою локальную сеть с Интернет (и поддерживает DHCP и DNS

серверы). Третий nameserver предоставляется мне сервис-провайдером. Кроме того, я установил 3-х секундное время ожидания (timeout) для каждого сервера имен, вместо 5 секунд, установленных по умолчанию.

Клиентские утилиты DNS

BIND 9 поставляется с четырьмя основными клиентскими утилитами. Три из них -- `dig`, `nslookup` и `host` -- выполняют сходные функции, более или менее отличающиеся деталями. Все три утилиты представляют из себя приложения, выполняемые в командной строке и посылающие запросы в DNS resolver. В сущности они делают то, что другие клиентские приложения делают на внутреннем уровне, но выводя результаты своей деятельности на STDOUT. Наиболее мощным средством из описанных выше является dig, поскольку предоставляет наиболее широкий набор опций для задания запросов и конфигурирования формата вывода полученной информации.

Эти утилиты чаще всего используются для определения IP адреса из символьного доменного имени, но вы также можете сделать и обратный запрос или запросить другие типы записей, кроме записей типа "A". Например, команда `host -t MX gnosis.cx` покажет вам почтовые сервера домена gnosis.cx. Несколько полезных примеров:

Листинг 2. Запрос при помощи host о google.com

```
$ host google.com
google.com has address 72.14.207.99
google.com has address 64.233.187.99
```

Листинг 3. Запрос при помощи host о записи MX для gnosis.cx

```
$ host -t MX gnosis.cx
gnosis.cx mail is handled by 10 mail.gnosis.cx.
```

Для утилиты `nslookup`:

Листинг 4. nslookup использует сервер, установленный по умолчанию (на локальной машине)

```
$ nslookup gnosis.cx
Server:         0.0.0.0
Address:        0.0.0.0#53

Non-authoritative answer:
Name:   gnosis.cx
Address: 64.41.64.172
```

Обратный запрос с использованием утилиты `dig` и с указанием другого сервера имен:

Листинг 5. Обратный запрос через dig на иной сервер, чем установленный по умолчанию

```
$ dig @192.168.2.2 -x 64.233.187.99

; <<>> DiG 9.2.4 <<>> @192.168.2.2 -x 64.233.187.99
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 3950
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;99.187.233.64.in-addr.arpa.    IN  PTR

;; AUTHORITY SECTION:
187.233.64.in-addr.arpa. 2613   IN  SOA  ns1.google.com. dns-
admin.google.com.
2004041601 21600 3600 1038800 86400

;; Query time: 1 msec
;; SERVER: 192.168.2.2#53(192.168.2.2)
;; WHEN: Thu Nov 10 02:00:27 2005
;; MSG SIZE  rcvd: 104
```

Последняя утилита BIND 9, о которой надо иметь представление, -- это `rndc`, утилита для управления сервером имен. Она является расширением утилиты `ndc`, поставлявшейся с предыдущими версиями BIND. Если `rndc` вызывается из командной строки без опций и аргументов, она выводит короткую справку о поддерживаемых командах. См. страницы системного руководства `man` для получения полной информации об использовании `rndc`.

Задание базовой конфигурации BIND и запуск сервера имен

Конфигурирование BIND

Запуск демона named в качестве DNS сервера возможен в одном из трех режимов: master, slave и caching-only. Демон named управляется своими конфигурационными файлами, в первую очередь /etc/bind/named.conf, для определения режима запуска.

В master моде сервер named работает как авторитетный источник всей информации об этой зоне. Информация о домене, предоставляемая сервером, размещается в файле на локальном диске, который можно модифицировать или обновить вручную. Каждая зона DNS должна иметь единственный мастер-сервер.

В slave моде сервер named передает информацию, предоставленную мастер-сервером зоны. Технически, сервер, поддерживающий несколько зон, может играть роль master'a для одной зоны и slave для другой, но обычно в локальной сети присутствует единственная иерархия из master, slave и caching-only серверов.

Сервер slave переносит полную информацию о зоне с мастер-сервера в локальные файлы, так что информация, предоставляемая slave сервером, по-прежнему остается авторитетной.

В режиме caching-only сервер named не поддерживает файлы зон. Каждый запрос перенаправляется на какой-то другой сервер имен для получения первичной информации, но затем полученная информация кешируется. Однако новые запросы требуют передачи запроса далее по сети. Caching-only сервера чаще всего используются на локальных машинах, где клиентские приложения часто посылают запросы в службу имен без генерации при этом сетевого трафика.

В конфигурации /etc/resolv.conf, показанной мной ранее в Листинге 1, 0.0.0.0 -- это caching-only сервер, 192.168.2.1 -- slave сервер и 151.203.0.84 -- как master сервер. Вы не можете определить этого, просто глядя на порядок их следования или на их IP адреса, но использование псевдо-IP адреса локальной машины дает основание полагать, что это caching-only сервер.

Создание named.conf

Есть несколько стандартных моментов, которые должны быть включены в каждый файл /etc/bind/named.conf. Это начальная директива `options`, задающая некоторую базовую информацию. Затем несколько директив `zone`, задающих конфигурацию для обработки различных зонных запросов. Домены, поименованные в директивах `zone` так, что их название начинается с IP адресов, задающих начальные цифры диапазона IP адресов, обозначают "обратные" зоны. Символьные имена определяют зоны, а кроме того, позволяют дальнейшее определение поддоменов.

Файлы named.conf (и другие конфигурационные файлы BIND) следуют соглашениям по форматированию языка C для больших фрагментов текста. Могут использоваться как C-подобные блочные комментарии (`/* comment */`) и принятые в C++ строчные (`// comment`), так и строчные комментарии shell (`# comment`). Директивы завершаются точками с запятой и заключаются в фигурные скобки.

Для начала несколько обычных опций. Мой локальный файл /etc/bind/named.conf начинается с:

Листинг 6. Мой локальный named.conf начинается так

```
include "/etc/bind/named.conf.options";
```

Но вы можете также вставлять непосредственно директиву `options`:

Листинг 7. Задание опций в named.conf

```
options {
    directory "/var/bind";
    forwarders { 192.168.2.1; 192.168.3.1};
    // forward only;
}
```

Эти установки указывают на то, что файлы, указанные без полного пути, будут искаться в указанном каталоге; кроме того, BIND в первую очередь обращается к 192.168.2.1 и 192.168.3.1 для незакэшированных результатов. Директива `forward only` (здесь закомментированная) говорит о том, что запросы выполняются на эти сервера имен, а не запрашиваются корневые сервера в Интернете.

Специальная директива `zone` должна помещаться ранее других в файлах named.conf:

Листинг 8. Особая зона (Hint zone) корневых серверов

```
zone "." {
    type hint;
    file "/etc/bind/db.root";
};
```

Содержимое `db.root` (иногда называемого `named.ca` для "certifying authority") носит достаточно специальный характер. Оно представляет из себя описание набора канонических корневых серверов, собственно регистрирующих домены. Этот файл меняется достаточно редко, но вы всегда может получить его последнюю официальную версию с ftp.rs.internic.net. Это не тот файл, который будет править обычный администратор.

За корневой особой зоной в named.conf должны размещаться master и/или slave зоны. Например, для локального loopback'а:

Листинг 9. Конфигурация loopback зоны

```
zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};
```

Что более интересно, named может выступать как master домена (в том числе предоставляя и обратные записи):

Листинг 10. Конфигурация внешней зоны

```
zone "example.com" {
    type master;
    file "example.com.hosts";  // file relative to /var/bind
};
// Reverse lookup for 64.41.* IP addresses (backward IP address)
zone "41.64.in-addr.arpa" {
    type master;
    file "41.64.rev";
};
```

Для slave конфигурации вместо этого вы может использовать:

Листинг 11. Конфигурация внешней зоны (slave)
```
zone "example.com" {
    type slave;
    file "example.com.hosts";  // file relative to /var/bind
    masters { 192.168.2.1; };
};
// Reverse lookup for 64.41.* IP addresses (backward IP address)
zone "41.64.in-addr.arpa" {
    type slave;
    file "41.64.rev";
    masters { 192.168.2.1; };
};
```

Другие конфигурационные файлы

Файл named.conf ссылается на ряд других конфигурационных файлов через директиву file. Они могут различаться в зависимости от конкретной установки, но обычно они будут содержать какие-то записи, определенные в RFC 1033 (Domain Administrators Operations Guide; см. в разделе Ресурсы). Стандартные записи:

SOA
: Start of authority (Начало полномочий). Параметры, действующие на всю зону.

NS
: Nameserver (Сервер имен). Доменный сервер имен.

A
: Address (Адрес). Имя хоста или IP адрес.

PTR
: Pointer (Указатель). IP адрес или имя хоста.

MX
: Mail exchange (Почтовая станция). Где обслуживается почта домена.

CNAME
: Canonical name (Каноническое имя). Псевдоним хоста.

TXT
: Text (Текст). Хранит произвольные значения.

Формат записи следующий: `<name> <time-to-live> IN <type> <data>`.

Имя и "время жизни" (time-to-live) опциональны, по умолчанию используются последние значения. Символьная строка IN обозначает Internet и частенько используется в практике. Файлы с записями ресурсов могут также содержать директивы, начинающиеся со знака доллара. Наверное, наболее часто в `$TTL`, устанавливающем умолчание для времени жизни. Например, какой нибудь тривиальный файл записей для `127.*` localhost'a выглядит так:

Листинг 12. Простейший файл записей
```
# cat /etc/bind/db.127
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
```

```
                              1     ; Serial
                         604800     ; Refresh
                          86400     ; Retry
                        2419200     ; Expire
                         604800 )   ; Negative Cache TTL
;
@       IN      NS      localhost.
1.0.0   IN      PTR     localhost.
```

Другие директивы -- это `$ORIGIN`, устанавливающая имя домена, используемое для дописывания любого относитльного имени домена; `$INCLUDE`, которая подгружает внешний файл; и `$GENERATE`, создающая серию записей, лежащих в определенном диапазоне IP адресов.

Создание и поддержание DNS зон

Файлы обратных зон

Файлы обратных зон (часто имеющих расширение .rev) -- содержащие отображение IP адресов данной зоны на их символьные имена. Например, у вас может быть файл /var/bind/41.64.rev, который содержит:

Листинг 13. Файл обратной зоны для 64.41.*

```
$TTL 86400
; IP address to hostname
@       IN      SOA     example.com.  mail.example.com. (
                              2001061401 ; Serial
                              21600      ; Refresh
                              1800       ; Retry
                              604800     ; Expire
                              900 )      ; Negative cach TTL

                IN      NS    ns1.example.com.
                IN      NS    ns2.example.com.
; Define names for 64.41.2.1, 64.41.2.2, etc.
1.2             IN      PTR   foo.example.com.
2.2             IN      PTR   bar.example.com.
3.2             IN      PTR   baz.example.com.
```

Файлы прямых зон

Файлы прямых зон (обычно называемые как domain.hosts) -- содержащие основные записи "A" отображающие символьные имена в IP адреса. Например, у вас может быть файл /var/bind/example.com.hosts, содержащий следующее:

Листинг 14. Прямая зона для example.com

```
$TTL 86400
; Hostname to IP address
@       IN      SOA     example.com.  mail.example.com. (
                                2001061401   ; Serial
                                21600        ; Refresh
                                1800         ; Retry
                                604800       ; Expire
                                900 )        ; Negative cach TTL

                IN      NS    ns1.example.com.
                IN      NS    ns2.example.com.
localhost       IN      A     127.0.0.1
foo             IN      A     64.41.2.1
www             IN      CNAME foo.example.com
bar             IN      A     64.41.2.2
bar             IN      A     64.41.2.3
```

Защита DNS сервера

Защита DNS сервера

Как и для многих других сервисов, запуск BIND в так называемом chroot jail окружении является хорошей идеей. Это ограничивает доступ BIND к другим файлам или системным ресурсам, при взломе BIND или наличии в нем ошибок. Более детальную информацию о запуске BIND в chroot окружении можно найти в "Chroot-BIND HOWTO" (см. в разделе Ресурсы).

Суть этой процедуры заключается в том, что запуск BIND под суперпользователем root или даже под традиционным спец. пользователем типа "nobody". Обычно для запуска BIND создается пользователь "named". Файлы, используемые этим спец. пользователем, помещаются в локальный каталог, например /chroot/named/, и соответствующие подкаталоги.

BIND 9 предоставляет более четкую поддержку для chroot, чем BIND 8; для ее включения достаточно обычной сборки без специальных опций или установок в Makefile.

DNSSEC

Кроме проведения работ по общему повышению защищенности машины, на которой работает BIND, также возможно обеспечить гарантированную защиту в рамках самого протокола DNS. DNS Security Extensions (DNSSEC) представляет из себя набор расширений DNS, обеспечивающих аутентификацию и целостность.

DNS базируется на UDP в большей степени, чем на TCP, а стало быть, не имеет механизма верификации источника пакета. Другими словами, посылающие запрос к DNS могут получить в ответ ложные данные, например перенаправляющие

соединение на хост взломщика. Добавлением криптографических Transactional Signatures (TSIG) в DNS запросы DNSSEC может предупредить подмену (spoofing) DNS ответов. На каждом сервере BIND 9, который должен работать в защищенном режиме, должен быть включен DNSSEC. Расширенный протокол, с другой стороны, обладает обратной совместимостью. В первую очередь DNS сервера, которые желают обмениваться данными в защищенном режиме, должны каким-то образом сгенерировать пару ключей. Это работает так же, как и SSH ключи для хоста и сервера. Например:

Листинг 15. Генерация ключей DNSSEC

```
dnssec-keygen -r /dev/urandom -a HMAC-MD5 -b 128 -n HOST \
primary-secondary.my.dom
# ls Kprimary-secondary.my.dom.*
Kprimary-secondary.my.dom.+157+46713.key
Kprimary-secondary.my.dom.+157+46713.private
```

Как подсказывают нам имена файлов, генерируются открытый (public) и закрытый (private) ключи для конфигурируемого хоста и публичный ключ для распространения по другим серверам. Хорошим введением в DNSSEC является "The Basics of DNSSEC" из O'Reilly Network (см. в разделе Ресурсы).