Unbound - настройка кеширующего DNS-сервера

Опубликовано muff в Втр, 2014-03-04 03:25

В релизе **FreeBSD 10.0 DNS**-сервер **BIND** заменен на связку из кеширующего **DNS-**сервера **Unbound** и библиотеки **LDNS**. Разбираясь с нововведениями, решил заодно ознакомиться и с настройкой **Unbound**.

Unbound распространяется под лицензией **BSD**, имеет модульную структуру и поддерживает работу резолвера в рекурсивном и кэширующем режиме. Во время работы сервера, кеш целиком распологается в памяти. Также имеется возможность проверки валидности **DNSSEC** -сигнатур, асинхронных запросов и библиотеки для интеграции кода резолвера в пользовательские приложения (**stub-resolvers**). Вначале прототип сервера был написан на языке **Java**, после чего был переписан на языке Си, что позволило значительно увеличить его производительность. По сравнению с **BIND**, стоит отметить скромные размеры и высокую производительность.

Итак, приступим к настройке... Поиск примера конфигурационного файла в только установленной системе результатов не дал... Однако на помощь пришла утилита **unbound-checkconf**:

```
# unbound-checkconf
[1393875321] unbound-checkconf[1362:0] error: Could not open /var/unbound/unbound.conf: No such file or directory
```

Конфигурационный файл **unbound.conf** должен находиться в каталоге **/var/unbound**. С доступными опциями решил ознакомиться на страницах руководства **man unbound.conf**. Как оказалось, пример конфигурационного файла был предложен именно там. Попробуем собрать небольшой файл конфигурации, отталкиваясь от предложеного примера и доступных опций.

```
server:# Уровень логирования - 0 (только ошибки)verbosity: 0
# Порт, на котором "слушать" запросыport: 53
# Описываем интерфейсы, на которых будем "слушать" запросы
interface: 127.0.0.1interface: 10.3.159.254# Указываем исходящий интерфейс
outgoing-interface: 10.12.59.30#  Указываем сети, чьи запросы будем обрабатывать
access-control: 10.3.159.0/24 allow
# разрешаем  ip4 tcp/udp и запрещаем поддержку ipv6
do-ip4: yesdo-ip6: nodo-udp: yesdo-tcp: yes
# От чьего имени работает daemon unboundusername: unbound
# Указываем лог-файл и отключаем использование syslog
logfile: "unbound.log"use-syslog: no# Указываем путь к pid-файлу
pidfile: "/var/run/local_unbound.pid"# "Прячем" версию софтahide-version: yes
```

После создания конфигурационного файла проверим конфигурацию с помощью уже известной утилиты **unbound-checkconf**:

```
# unbound-checkconf
unbound-checkconf: no errors in /var/unbound/unbound.conf
```

Кажется все в порядке. Добавим загрузку демона при старте системы:

```
# echo '# DNS server' >> /etc/rc.conf
# echo 'local_unbound_enable="YES"' >> /etc/rc.conf
```

Кажется ничего не пропустили... Даем команду на запуск:

```
# sh /etc/rc.d/local_unbound start
Starting local_unbound.
```

Проверим, запустился ли процесс:

```
# ps -ax | grep unbound | grep -v grep
2340  -  Is     0:00,03 /usr/sbin/unbound -c/var/unbound/unbound.conf
```

Демон запущен. Проверим, обрабатывает ли он запросы:

```
# drill @127.0.0.1 muff.kiev.ua
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 57961
;; flags: qr rd ra ; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;; muff.kiev.ua.        IN      A

;; ANSWER SECTION:
muff.kiev.ua.   3427    IN      A       91.196.100.50

;; AUTHORITY SECTION:

;; ADDITIONAL SECTION:

;; Query time: 0 msec
;; SERVER: 127.0.0.1
;; WHEN: Tue Mar  4 01:43:24 2014
;; MSG SIZE  rcvd: 46
```

Клиентские запросы обрабатываются. Отлично, продолжим настройку...

Следующий шаг - настройка утилиты **unbound-control** на работу с **unbound**. Для этого воспользуемся прилагаемой утилитой **unbound-control-setup**:

```
# unbound-control-setup


setup in directory /etc/unboundgenerating unbound_server.keyGenerating RSA private k
ey, 1536 bit long modulus...............++++.......................................
.....................++++e is 65537 (0x10001)generating unbound_control.keyGeneratin
g RSA private key, 1536 bit long modulus.......................................
....................++++.......................................................
.......................++++e is 65537 (0x10001)create unbound_server.pem (self sign
ed certificate)create unbound_control.pem (signed client certificate)Signature oksub
ject=/CN=unbound-controlGetting CA Private KeySetup success. Certificates created. E
nable in unbound.conf file to use
```

После этого необходимо добавить следующий блок в конфигурационный файл **unbound.conf**:

```
remote-control:
control-enable: yes
control-interface: 127.0.0.1
control-port: 953
server-key-file: "unbound_server.key"
server-cert-file: "unbound_server.pem"
control-key-file: "unbound_control.key"
control-cert-file: "unbound_control.pem"
```

Чтобы изменения вступили в силу, необходимо перезапустить **unbound**:

```
# sh /etc/rc.d/local_unbound restart
```

Чтобы ознакомиться с возможностями утилиты **unbound-control**, достаточно вызвать ее без ключей, либо же с ключем **-h**:

```
# unbound-control -h


Usage:  unbound-control [options] command        Remote control utility for unbound
server.Options:  -c file       config file, default is /var/unbound/unbound.conf  -s
 ip[@port]  server address, if omitted config is used.  -q            quiet (don't p
rint anything if it works ok).  -h            show this usage help.Commands:  start
                        start server; runs unbound(8)  stop
 stops the server  reload                        reloads the server
               (this flushes data, stats, requestlist)  stats
  print statistics  stats_noreset                 peek at statistics  status
                display status of server  verbosity <number>            change loggi
ng detail  log_reopen                     close and open the logfile  local_zone <nam
e> <type>      add new local zone  local_zone_remove <name>      remove local zone a
nd its contents  local_data <RR data...>       add local data, for example
                       local_data www.example.com [1]
 A 192.0.2.1  local_data_remove <name>     remove local RR data from name  dump_cache
      print cache to stdout  load_cache               load cache from stdin  lookup <name>
print nameservers for name  flush <name>              flushes common types for name from cache
                    types:  A, AAAA, MX, PTR, NS,                           SOA, CNAME, DNAME, SRV
, NAPTR  flush_type <name> <type>     flush name, type from cache  flush_zone <name>           f
lush everything at or under name                        from rr and dnssec caches  flush_bogus
       flush all bogus data  flush_stats                flush statistics, make zero  flush_requestlist
drop queries that are worked on  dump_requestlist           show what is worked on  flush_infra [all |
 ip]       remove ping, edns for one IP or all  dump_infra                  show ping and edns entries  set_
option opt: val          set option to value, no reload  get_option opt              get option value  list_st
ubs                list stub-zones and root hints in use  list_forwards              list forward-zones in us
e  list_local_zones            list local-zones in use  list_local_data             list local-data RRs in use  fo
rward_add [+i] zone addr..  add forward-zone with servers  forward_remove [+i] zone      remove for
ward zone  stub_add [+ip] zone addr..    add stub-zone with servers  stub_remove [+i] zone         re
move stub zone              +i            also do dnssec insecure point              +p            set stub to
use priming  forward [off | addr ...]      without arg show forward setup                          or off to t
urn off root forwarding                          or give list of ip addressesVersion 1.4.20BSD licensed, se
e LICENSE in source package for details.Report bugs to unbound-bugs [at] nlnetlabs [dot] nl
```

Здесь, благодаря комментариям, все интуитивно понятно

Проверим работоспособность утилиты, отправив какую либо команду в **unbound-control**, например **reload**:

```
# unbound-control reload
ok
```

Поддерживается возможность управления удаленным сервером **unbound**, используя локальную версию утилиты **unbound-control**. Для этого необходимо, чтобы на машине, с которой выполняются команды, публичные **pem**-ключи удаленного сервера. Также необходимо, чтобы эти ключи были прописаны в конфигурационном файле **unbound.conf**.

Также **unbound** поддерживает использование **DNSSEC**. Рассматривать что такое **DNSSEC** и принципы его работы не вижу смысла - информации в Сети предостаточно. Остановимся непосредственно на настройке **unbound** для поддержки **DNSSEC**.

Для включения поддержки **DNSSEC**, в конфигурационный файл **unbound.conf** (в секцию **server**) необходимо добавить следующие строки:

```
module-config: "validator iterator"
auto-trust-anchor-file: "/var/unbound/root.key"
```

После внесения изменений необходимо перезапустить **unbound**, чтобы изменения вступили в силу:

```
# unbound-control reload
ok
```

Проверим поддержку **DNSSEC**:

```
# drill -D example.com @127.0.0.1

;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 30826
;; flags: qr rd ra ad ; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 0
;; QUESTION SECTION:
;; example.com. IN      A

;; ANSWER SECTION:
example.com.    86072   IN      A       93.184.216.119
example.com.    86072   IN      RRSIG   A 8 2 86400 20140319191725 20140312122744 6439
example.com. VZWTz7/E6engcARhn1KO00NTA+BSB8NuOOdaBjyMtu3qx5AIlmkNR4ifUHUsOS4V1K5
yC25GnmDa7lIE1FwFEiQvISUmD41bRX9kJuHLt9JhRj5hrb7L+9aqeeTFa2jSRk59WIaJhbqcKYFTdm8BG
sGdx6lHx67X4W+nN7BRUUo=

;; AUTHORITY SECTION:
example.com.    172472  IN      NS      b.iana-servers.net.
example.com.    172472  IN      NS      a.iana-servers.net.
example.com.    172472  IN      RRSIG   NS 8 2 172800 20140319163437 20140312122744 6439
example.com. IOIONMwaAc6pGLYh5yOL7SHDNY34vG5W/O8chCTHlIkgcVij2KCXtrrVCUOVyne1nhUgf
EZHVvRU0VwmXecqVSz5Oe6iWPyZ7v8CBfgFjvS/oNlJ+8uBCAopxiIWv4EnCCsu1RUXoIBD2N2kwj3CF1
Ba3qGDjkGMTzN5pblcuck=

;; ADDITIONAL SECTION:

;; Query time: 0 msec
;; EDNS: version 0; flags: do ; udp: 4096
;; SERVER: 127.0.0.1
;; WHEN: Wed Mar 12 16:37:19 2014
;; MSG SIZE  rcvd: 446
```

Ну и на всякий случай - проверим поддержку DNSSEC непосредственно в браузере. С целью проверки установим для **Firefox** плагин **DNSSEC Validator**. Результат - на картинке внизу:

[2]

Базовая настройка **Unbound** закончена, однако есть еще много опций конфигурации, ознакомиться с которыми можно в **man unbound.conf**.

Источник (получено *2025-07-15 23:23*):
http://muff.kiev.ua/content/unbound-nastroika-keshiruyushchego-dns-servera

Ссылки:
[1] http://www.example.com
[2] http://muff.kiev.ua/files/imagepicker/1/unbound_dnssec.png