Одним из важных сервисов, обеспечивающих функционирование современного интернета является сервис по преобразованию имени сайта в ip адрес. Настройкой реализации сервиса DNS мы займемся в этой статье на примере настройки Bind 9 (named) на сервере под управлением CentOS 7. Мы подготовим минимально необходимый базовый функционал и заглянем немного глубже в настройки логирования.

Содержание:

Что такое DNS сервер BIND

Bind - самая распространенная на текущий день реализация ДНС сервера, которая обеспечивает преобразование IP адресов в dns-имена и наоборот. Его также называют named, например в Freebsd. Судя по информации из Википедии, сейчас 10 из 13 корневых ДНС серверов интернета работают на bind. Он установлен из коробки практически во всех linux дистрибутивах. Я рассмотрю его установку на сервер CentOS 7.

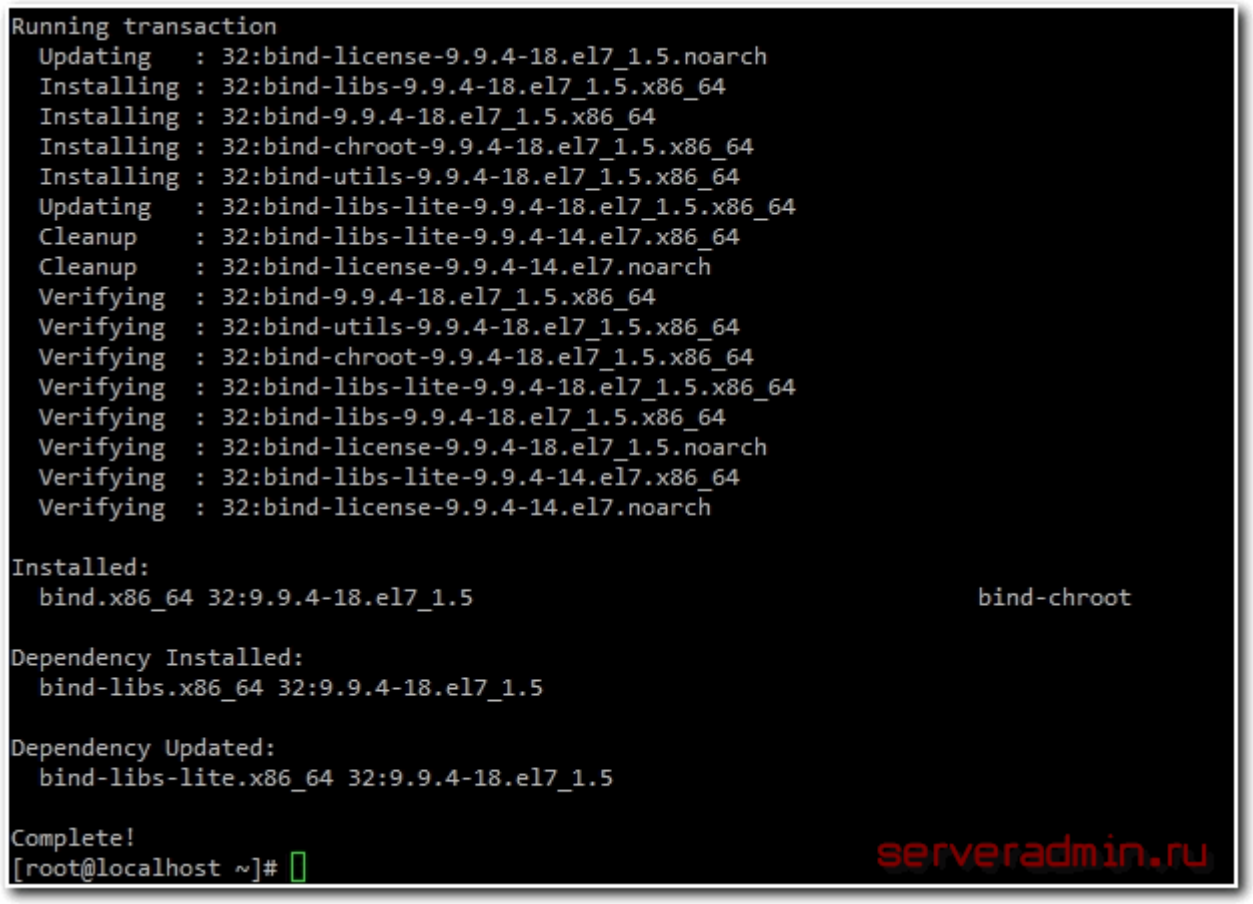
Устанавливаем Bind 9 (named) в CentOS 7

Первым делом проверим, установлен ли у нас днс сервер в системе:

```
# rpm -qa bind*
 bind-libs-lite-9.9.4-14.el7.x86_64
 bind-license-9.9.4-14.el7.noarch
```

У меня не установлен, так как во время инсталляции centos выбрал минимальный пакет программ. Сервер имен у нас будет работать в chroot окружении, так что устанавливаем соответствующие пакеты:

```
# yum -y install bind bind-utils bind-chroot
```



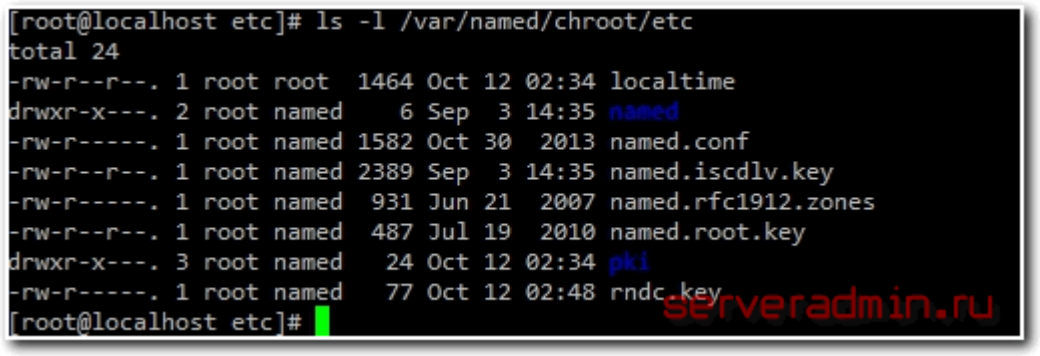
Еще раз обращаю внимание, что мы будем использовать bind в **chroot** среде для увеличения безопасности. Это накладывает определенные особенности в настройке и управлении сервером. Нужно быть внимательным в этих мелочах. Итак, запускаем bind:

```
# systemctl start named-chroot
```

```
# systemctl enable named-chroot
ln -s '/usr/lib/systemd/system/named-chroot.service' '/etc/systemd/system/multi-user.target.wants/named-chroot.service'
```

Проверяем содержимое chroot каталога:

```
# ls -l /var/named/chroot/etc
```



Все в порядке, сервер запустился, необходимые файлы созданы, все готово для настройки. Займемся ей.

Настраиваем DNS сервер в CentOS 7

Файл конфигурации нашего сервера располагается по адресу /var/named/chroot/etc/named.conf. Открываем его и приводим к следующему виду:

```
# mcedit /var/named/chroot/etc/named.conf
```

```
options {
listen-on port 53 { any; };
listen-on-v6 port 53 { none; };
directory "/var/named";
dump-file "/var/named/data/cache_dump.db";
allow-query { 127.0.0.1; 192.168.7.0/24; };
recursion yes;
allow-recursion { 127.0.0.1; 192.168.7.0/24; };
forwarders { 8.8.8.8; };
version "DNS Server";
managed-keys-directory "/var/named/dynamic";
pid-file "/run/named/named.pid";
session-keyfile "/run/named/session.key";
dnssec-enable no;
dnssec-validation no;
};

zone "." IN {
type hint;
file "named.ca";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";

logging {
channel default_file {
file "/var/log/named/default.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};
category default { default_file; };
```

```
};
```

Эта конфигурация обеспечит работу обычного кэширующего сервера в локальной сети. Комментарии к некоторым параметрам:

listen-on-v6 port 53 { none; };	Отключили работу на интерфейсе ipv6.
allow-query { 127.0.0.1; 192.168.7.0/24; };	Разрешаем обычные запросы только из локальной сети.
allow-recursion { 127.0.0.1; 192.168.7.0/24; };	Разрешаем рекурсивные запросы только из локальной сети.
forwarders { 8.8.8.8; };	Перенаправляем запросы, которые сами не резолвим, на днс сервер гугла. У меня указан он просто для примера. Тут лучше всего указать сначала ДНС серверы провайдера.
version "DNS Server";	Скрываем версию бинда, вместо этого выводим указанную строку.

> Не забудьте отредактировать правила фаервола для корректной работы DNS сервера - открыть 53 порт UDP для работы кэширующего сервера, который мы сейчас настроили, и 53 порт TCP для пересылки зон, о которых речь пойдет дальше

Теперь создадим папку для логов. Не забываем, что мы работаем в chroot окружении:

```
# cd /var/named/chroot/var/log && mkdir named && chown named. named
```

Поддержка собственной зоны

Допустим, нам необходимо в нашем named разместить собственную зону site1.ru. Первым делом создаем файл зоны, которую будет обслуживать dns сервер:

```
# mcedit /var/named/chroot/var/named/site1.ru.zone
```

```
$TTL 86400
```

```
@       IN     SOA     site1.ru. site1.ru.local. (
                       2015092502
                       43200
                       3600
                       3600000
                       2592000 )

        IN NS ns1.site1.ru.
        IN NS ns2.site1.ru.
        IN A 192.168.7.254
        IN MX 10 mx.site1.ru.

gate    IN A 192.168.7.254
mx      IN A 192.168.7.250
ns1     IN A 192.168.7.235
ns2     IN A 192.168.7.231
```

Описание синтаксиса файлов зон достаточно хорошо освещено в интернете, не хочется подробно на этом останавливаться. При желание каждый сам сможет посмотреть, если у него возникнет необходимость настроить поддержку собственной зоны.

Выставляем необходимые права:

```
# chown root:named /var/named/chroot/var/named/site1.ru.zone
# chmod 0640 /var/named/chroot/var/named/site1.ru.zone
```

Дальше подключаем файл зоны в конфигурационном файле bind - /var/named/chroot/etc/named.conf:

```
zone "site1.ru" {
 type master;
 file "site1.ru.zone";
```

```
};
```

Перечитываем конфигурацию named с помощью **rndc**:

```
# rndc reconfig
```

Добавление в bind slave zone

Если вы хотите на своем сервере держать копию какой-то зоны, взятой с другого dns сервера, то добавьте следующие настройки в конфиг.

```
zone "site.ru" IN {
 type slave;
 masters { 10.1.3.4; };
 file "site.ru.zone";
};
```

10.1.3.4 - ip адрес dns сервера, с которого мы берем зону. Не забудьте на нем разрешить передачу зоны на ваш dns сервер.

Чтобы сервер смог корректно сохранить файл со slave зоной, необходимо добавить разрешение на запись bind для директории */var/named/chroot/var/named*. По-умолчанию она имеет следующие права:

```
drwxrx--- 6 root named 164 Jan 6 06:06 named
```

Нужно добавить группе named разрешение на запись, чтобы стало вот так:

```
drwxrwx--- 6 root named 164 Jan 6 06:06 named
```

После этого можно перезапустить bind и проверить, что создался файл слейв зоны. С указанными выше настройками, он будет располагаться по адресу */var/named/chroot/var/named/site.ru.zone*. Если у bind не будет прав для создания файла, в логе вы получите ошибку:

```
dumping master file: tmp-7Swr6EZpcd: open: permission denied
```

Настройка логов в bind (named)

Гораздо интереснее и полезнее разобраться с подробным логированием работы сервера. Я долгое время поверхностно хватался за всякие рекомендации и куски примерных конфигов в интернете, пока в не решил разобраться сам с этой темой и не полез в оригинальный мануал.

Bind дает широкие возможности для ведения логов. Можно фиксировать практически все, что связано с работой сервера. Я сейчас на простых примерах покажу, как это работает.

Первым делом в конфигурации мы задаем канал, куда будут складываться логи по тем или иным событиям. Вот пример подобного канала:

```
channel general {
file "/var/log/named/general.log" versions 3 size 5m;
severity dynamic;
print-time yes;
```

Здесь указано название канала, которые мы придумываем сами - general, указан путь до файла, сказано, что хранить будем 3 версии лога размером не более 5 мегабайт. Параметр **severity** может принимать следующие значения:

Описание параметров severity

critical	Только критические ошибки.
error	Обычные ошибки и все что выше.
warning	Предупреждения и все, что выше.
notice	Уведомления и все, что выше.
info	Информационные сообщения и все что выше.

debug	Сообщения уровня debug и все, что выше. Уровни debug регулируются значениями 0, 1, 2, 3.
dynamic	То же, что и debug, только его уровень регулируется глобальной настройкой сервера.

Параметр **print-time** указывает на то, что в лог необходимо записывать время события. Помимо указанных мной настроек, в конфигурации канала могут быть добавлены следующие параметры:

- **print-severity** yes | no - указывает, писать или нет параметр severity в лог
- **print-category** yes | no - указывает писать или нет название категории логов

Я эти параметры не указал, так как по-умолчанию устанавливается значение **no**, которое лично меня устраивает.

Дальше необходимо указать категорию логов и в какой канал мы будем ее записывать:

```
category general { general; };
```

Категорий у днс сервера bind достаточно много. Вот мой перевод полного списка с описаниями:

Описание категорий логов в bind (named)

default	Сюда будут попадать события всех категорий из этой таблицы, если они не определены отдельно, за исключением категории queries, которую нужно включать специально. То есть если обозначить только категорию default, то в нее будут сыпаться события всех категорий.
general	Эта категория для всех логов, которые не включены ни в одну из перечисленных категорий.
database	Сообщения, относящиеся к хранению зон и кэшированию.
security	Подтверждение и отказ в выполнении запросов.
config	Все, что относится к чтению и выполнению файла конфигурация.
resolver	Разрешение имен, включая информацию о рекурсивных запросах, выполняемых от имени клиента кэширующим сервером.
xfer-in	Информация о получении зон.
xfer-out	Информация о передаче зон.
notify	Логирование операций протокола NOTIFY.
client	Выполнение клиентских запросов.

unmatched	Сообщения, которые named не смог отнести ни к одному классу или для которых не определено отображение.
network	Логирование сетевых операций.
update	Динамические апдейты.
update-security	Подтверждение или отклонение запросов на апдейт.
queries	Логирование запросов к ДНС серверу. Для включения этой категории необходимо отдельно задать параметр в конфигурации сервера. Это связано с тем, что эта категория генерирует очень много записей в лог файл, что может сказаться на производительности сервера.
query-errors	Ошибки запросов к серверу.
dispatch	Перенаправление входящих пакетов модулям сервера на обработку.
dnssec	Работа протоколов DNSSEC и TSIG.
lame-servers	Фиксируются ошибки, которые получает bind при обращении к удаленным серверам в попытке выполнить запрос на разрешение имени.
delegation-only	Логирование запросов, вернувших NXDOMAIN.
edns-disabled	Запросы, которые вынуждены использовать plain DNS из-за превышения timeouts.
RPZ	Все операции, связанные с выполнение Response Policy Zone (RPZ).
rate-limit	Операции связанные с одним или несколькими rate-limit statements в options или view.

Таким образом, чтобы вывести все категории логов в отдельные файлы, необходимо в конфиг named добавить следующую конструкцию:

```
logging {

channel default {
file "/var/log/named/default.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel general {
file "/var/log/named/general.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};
```

```
channel database {
file "/var/log/named/database.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel security {
file "/var/log/named/security.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel config {
file "/var/log/named/config.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel resolver {
file "/var/log/named/resolver.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel xfer-in {
file "/var/log/named/xfer-in.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel xfer-out {
file "/var/log/named/xfer-out.log" versions 3 size 5m;
severity dynamic;
```

```
print-time yes;
};

channel notify {
file "/var/log/named/notify.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel client {
file "/var/log/named/client.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel unmatched {
file "/var/log/named/unmatched.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel network {
file "/var/log/named/network.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel update {
file "/var/log/named/update.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};
```

```
channel update-security {
file "/var/log/named/update-security.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel queries {
file "/var/log/named/queries.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel query-errors {
file "/var/log/named/query-errors.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel dispatch {
file "/var/log/named/dispatch.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel dnssec {
file "/var/log/named/dnssec.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel lame-servers {
file "/var/log/named/lame-servers.log" versions 3 size 5m;
severity dynamic;
```

```
print-time yes;
};

channel delegation-only {
file "/var/log/named/delegation-only.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel edns-disabled {
file "/var/log/named/edns-disabled.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel rpz {
file "/var/log/named/rpz.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

channel rate-limit {
file "/var/log/named/rate-limit.log" versions 3 size 5m;
severity dynamic;
print-time yes;
};

category default { default; };
category general { general; };
category database { database; };
category security { security; };
category config { config; };
category resolver { resolver; };
```

```
category xfer-in { xfer-in; };
category xfer-out { xfer-out; };
category notify { notify; };
category client { client; };
category unmatched { unmatched; };
category network { network; };
category update { update; };
category update-security { update-security; };
category queries { queries; };
category query-errors { query-errors; };
category dispatch { dispatch; };
category dnssec { dnssec; };
category lame-servers { lame-servers; };
category delegation-only { delegation-only; };
category edns-disabled { edns-disabled; };
category rpz { rpz; };
category rate-limit { rate-limit; };
};
```

Если мы хотим собирать все логи запросов из категории **queries**, то в раздел options файла конфигурации необходимо добавить параметр, который это разрешает:

```
querylog yes;
```

Перезапускаем bind:

```
# systemctl restart named-chroot.service
```

Проверка работы DNS Server

Первым делом пойдем в каталог с логами и проверим, что там у нас:

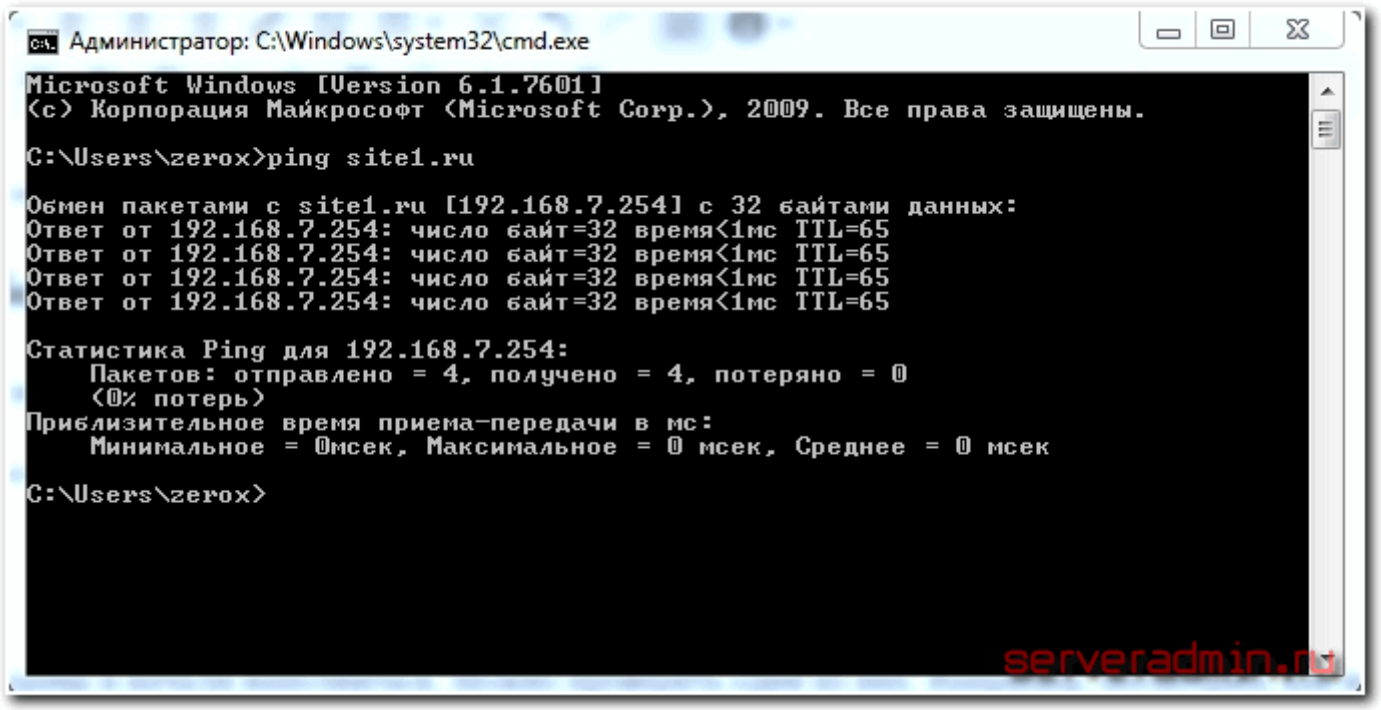
```
# cd /var/named/chroot/var/log/named
# ls -l
```

```
[root@localhost named]# ls -l
total 8
-rw-r--r-- 1 named named    0 Sep 26 19:19 client.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 config.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 database.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 default.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 delegation-only.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 dispatch.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 dnssec.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 edns-disabled.log
-rw-r--r-- 1 named named  185 Sep 26 19:19 general.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 lame-servers.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 network.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 notify.log
-rw-r--r-- 1 named named 1055 Sep 26 19:20 queries.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 query-errors.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 rate-limit.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 resolver.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 rpz.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 security.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 unmatched.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 update.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 update-security.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 xfer-in.log
-rw-r--r-- 1 named named    0 Sep 26 19:19 xfer-out.log
[root@localhost named]#
```


Все файлы журнала созданы и начали наполняться. Можно проверить один из них. Например, посмотрим, как наш сервер centos (192.168.7.246) логирует запросы пользователей. Попробуем с компьютера 192.168.7.254 (windows) выполнить nslookup yandex.ru и посмотрим как это отразится в лог файле:

```
26-Sep-2015 19:25:30.923 client 192.168.7.254#56374 (yandex.ru): query: yandex.ru IN A + (192.168.7.246)
26-Sep-2015 19:25:31.013 client 192.168.7.254#56375 (yandex.ru): query: yandex.ru IN AAAA + (192.168.7.246)
```

Теперь выполним ping site1.ru, чтобы проверить, как сервер поддерживает нашу зону:



Смотрим, что в логах:

```
26-Sep-2015 19:28:01.660 client 192.168.7.254#49816 (site1.ru): query: site1.ru IN A + (192.168.7.246)
```

Таким образом очень удобно отследить, куда лезет компьютер. Например, можно поднять временно dns сервер, включить лог запросов. В клиенте указать единственный днс сервер, который мы настроили. Дальше можно отслеживать, к примеру, куда лезет винда после загрузки без нашего ведома. Или откуда грузится реклама в скайпе. Все запросы будут аккуратно складываться в файл, который потом можно спокойно анализировать, а затем, к примеру,

настроить запрет сайтов на микротике.

Это все, что я хотел в данном материале рассказать. Тема настройки bind (named) достаточно обширная. Возможно я еще вернусь к ней.