



ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование



Меня хорошо видно && слышно?

Ставьте + , если все хорошо
Напишите в чат, если есть проблемы



НЕ ЗАБЫТЬ ВКЛЮЧИТЬ
ЗАПИСЬ!!!

Администратор Linux. Domain Name System.

Цель занятия

- Познакомиться с технологией Domain Name System (DNS)
- Успользовать утилиты для диагностики DNS
- Научиться управлять DNS-зонами

План занятия

- Теория DNS
 - Назначение
 - Понятия и определения;
 - Ресурсные записи;
 - Типы запросов;
 - Роли серверов;
- Практика
 - настройка сервера DNS;
 - добавляем файл зоны;
 - добавляем split view;

Domain Name Service

- Зачем имена?

```
# cat /etc/hosts
127.0.0.1
localhost localhost.localdomain localhost4 localhost4.localdomain
127.0.1.1 ns01 ns01
192.168.10.10 myhosts
```

- Что не так с файлом /etc/hosts?

Протокол использует для работы:

- 53/UDP
- 53/TCP (в особых случаях)

Использование DNS

Основное - сопоставление IP-адресов и DNS-имён. Бывают 2-х видов:

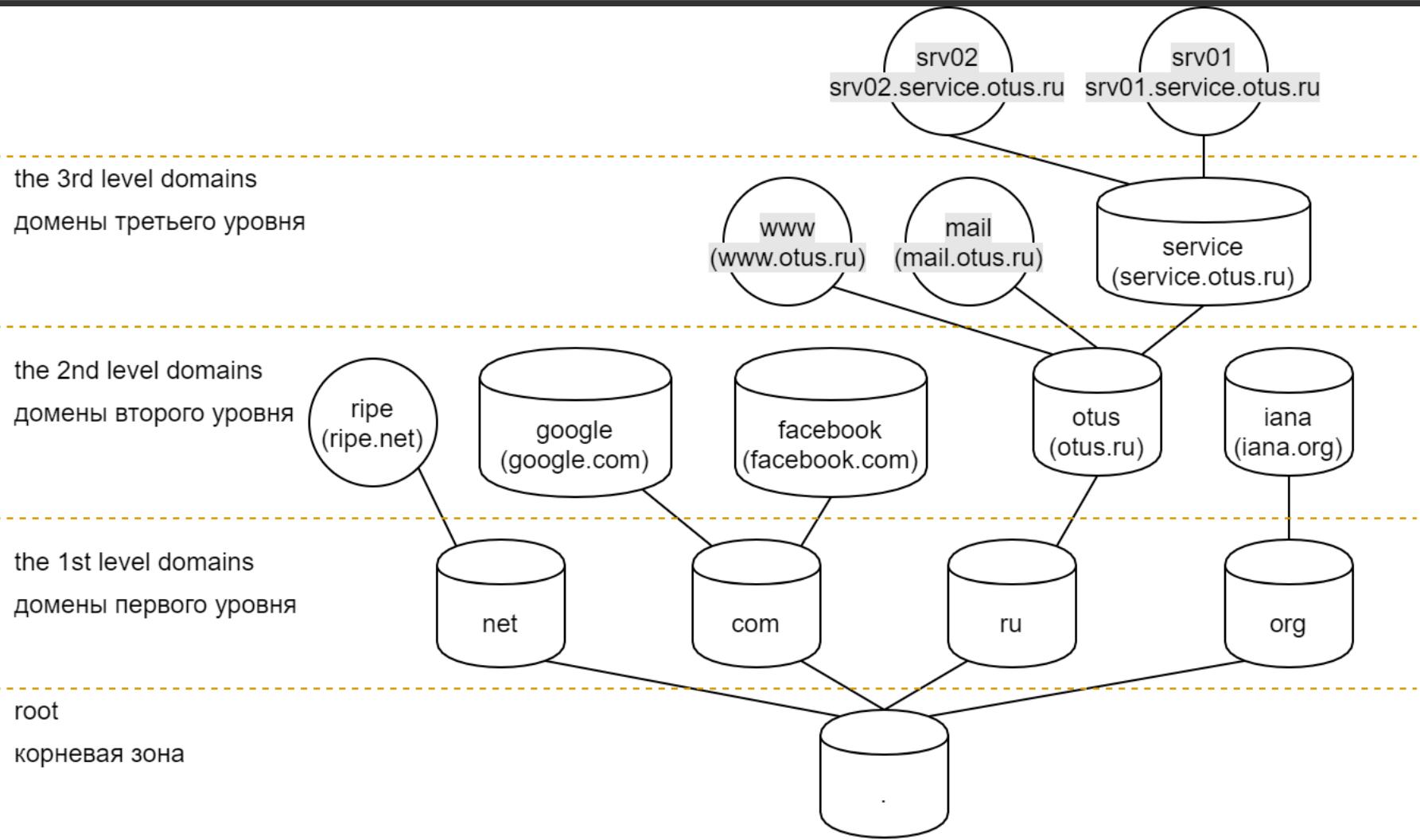
- прямое (имя в IP-адрес);
- обратное (IP-адрес в имя).

Также используется для хранения дополнительной информации:

- расположение сетевых сервисов;
- текстовой информации для служебных целей.

Может быть использован для балансировки нагрузки.

Структура



Fully Qualified Domain Name FQDN

- полностью указанное доменное имя, т.е. от корневого домена. Ключевой индикатор FQDN - точка в конце имени.

FQDN www.otus.ru. состоит из:

- домена 3-го уровня **www**, входящего в состав **otus.ru**;
- домена 2-го уровня **otus**, входящего в состав **ru**;
- домена 1-го уровня **ru**, входящего в состав корневого домена.

Корневой домен не имеет названия и обозначается в FQDN точкой - ":"

Клиент DNS. Резолвер

```
ping www.otus.ru  
curl www.otus.ru
```

Утилиты используют функции библиотеки glibc предоставляемые доступ к DNS

```
res_init, res_query, res_search, res_querydomain, res_mkquery, re  
man 3 resolver
```

Настройки в файле

```
/etc/resolv.conf  
nameserver 192.168.10.10  
search otus.ru  
options timeout:20
```

Детали: man 5 resolver

Клиент DNS. Утилиты

```
yum install bind-utils  
apt-get install bind9utils
```

Диагностика:

- host
- dig, delv
- nslookup

Утилиты позволяют делать запросы минуя стандартную библиотеку. Можно указать:

- IP адрес DNS сервера
- тип ресурсной записи

Узнаем IP адрес

Сравним стандартный вывод утилит. Опциями можно модифицировать кол-во информации

```
host www.otus.ru  
nslookup www.otus.ru  
dig www.otus.ru
```

Узнаем куда слать почту, адреса DNS серверов, текст и прочее.

```
dig NS www.otus.ru  
dig MX www.otus.ru  
dig SOA www.otus.ru  
dig TXT www.otus.ru
```

Ресурсные записи (RR)

Записи DNS обладают следующими атрибутами:

- имя;
- TTL (время жизни в кеше);
- класс;
- тип;
- значение (или массив значений).

www.otus.ru.	60	IN	A	1.2.4.5
mail	60	IN	MX	10 mx1.otus.ru.
			MX	20 mx1.otus.ru.

Типы записей

- **A** - адрес IPv4, соответствующий имени
- **AAAA** - адрес IPv6, соответствующий имени
- **CNAME** - имя, соответствующее имени (canonical name)
- **MX** - массив (приоритет и имя) почтовых серверов для домена
- **TXT** - текстовая информация
- **SOA** - ключевая запись домена (start of authority)
- **NS** - имя сервера имён для домена (nameserver)
- **PTR** - имя, соответствующее IP-адресу (pointer для in-addr.arpa и ip6.arpa)
- **SRV** - указание на расположение сервиса

Используются для: описания в файле зоны, отправке запроса, приходят в ответе на запрос

Как работает рекурсивный запрос?

Получим список корневых серверов

```
dig @d.root-servers.net . ns
```

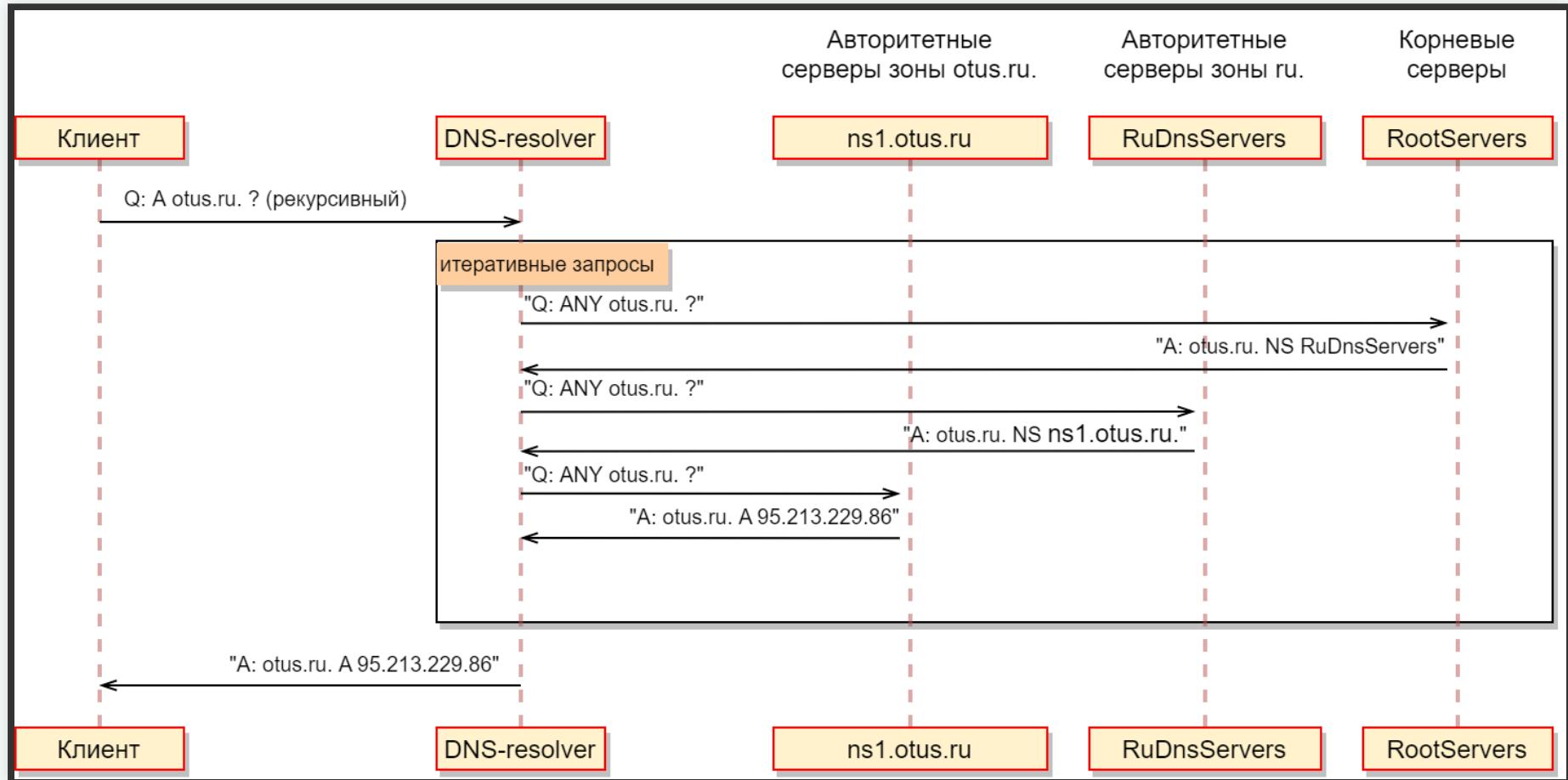
```
dig @d.root-servers.net www.otus.ru ns
dig @a.dns.ripn.net. www.otus.ru ns
dig @autumn.ns.cloudflare.com. www.otus.ru
```

Запрос рекурсивный или итеративный?

```
dig @autumn.ns.cloudflare.com. www.otus.ru\
| grep flags
dig www.otus.ru | grep flags
```

rd - Recursion Desired, ra - Recursion Available, aa - Authoritative Answer

Как работает рекурсивный запрос?



Типы серверов

Типы серверов (по свойствам и функциям):

- **главные (primary или master)** - авторитетный, хранят главную копию информации о зоне;
- **вторичные (secondary или slave)** - получают копию информации о зоне с главного или вторичного сервера и работают с ней;
- **кэширующие** - кэшируют ответы на запросы пользователя;
- **рекурсивные** - выполняют полный поиск по иерархии DNS;
- **нерекурсивные** - не выполняют полный поиск (не умеют или им запрещено).

Реализация серверов

- bind. Будем рассматривать BIND 9 версии
- powerdns
- unbound
- dnsmasq
- другие варианты.

Устанавливаем сервер

```
# CentOS
yum install bind bind-utils
# Конфиг
/etc/named.conf
```

```
# Ubuntu
apt-get install bind9 bind9utils dnsutils
# Конфиг
/etc/bind/named.conf
```

В конфигурационном файл задаем роль сервера, расположение файлов зоны

bind - acl

acl (access control list) - позволяют задать именованный список из сетей и/или TSIG-ключей. Впоследствии, список можно применять к оцпиям.

Формат раздела: acl "имя_сети" {ip; !ip; key; };

Описание особенностей вложенных ACL [тут](#).

bind - раздел options

- `directory /path/to/work/dir` - указывает абсолютный путь к рабочему каталогу сервера;
- `allow-query {список_ip}` - Разрешает ответы на запросы только из "список_ip". При отсутствии параметра сервер отвечает на все запросы;
- `allow-recursion {список_ip}` - На запросы из "список_ip" будут выполняться рекурсивные запросы. Для остальных - итеративные. Если не задан параметр, то сервер выполняет рекурсивные запросы для всех сетей;
- `allow-transfer {список_ip}` - Указывает "список_ip" серверов, которым разрешено брать зону с сервера (в основном тут указывают slave-серверы).

bind - раздел options

- `forwarders {ip порт, ip порт...}` - указывает адреса хостов и, если нужно порты, куда переадресовывать запросы (обычно тут указываются DNS провайдеров ISP);
- `forward ONLY|FIRST` - параметр `first` указывает, DNS-серверу пытаться разрешать имена с помощью DNS-серверов, указанных в параметре `forwarders`, и лишь в случае, если разрешить имя с помощью данных серверов не удалось, то будет осуществлять попытки разрешения имени самостоятельно;
- `notify YES|NO` - уведомлять slave-серверы об изменениях в зоне;
- `recursion YES|NO` - выполнять рекурсивные запросы, если просит клиент или нет, и выполнять только итеративные запросы. Если ответ найден в кэше, то возвращается из кэша.

bind - раздел zone

Формат раздела: zone {операторы_раздела_zone};

Операторы, которые наиболее часто используются:

- allow-update {список_ip} - кому разрешено динамически обновлять данную зону;
- file "имя_файла" - указывает путь файла параметров зоны (относительно directory или полный);
- masters {список_ip} - указывает список master-серверов (допустим только в подчиненных зонах);
- type "тип_зоны" - указывает тип зоны, описываемой в текущем разделе:
 - hint - указывает вспомогательную зону (информация о корневых серверах);
 - master - мастер сервер для текущей зоны;
 - slave - подчиненный сервер для текущей зоны;
 - forward - указывает зону переадресации.

TSIG (transaction signatures)

Для защиты от искажений и подделок ответов сервера, передачи зоны и обновлений зоны поддерживается использование расширения TSIG протокола DNS.

- Генерация ключа:
`dnssec-keygen -a HMAC-MD5 -b 128 -n HOST имя-ключа`
- Определение ключа:
`key имя-ключа { algorithm hmac-md5; secret "секретная-строка-в-base-64"; };`

Может использоваться для аутентификации и авторизации:

- view;
- server;
- controls (например, для rndc);
- acl и прочих списках.

Файл зоны (с ошибками)

```
ORIGIN domain.tld.
$TTL 3600
@ 600 IN SOA    ns1.domain.tld.    hostmaster.domain.tld. (
                    0 ; Serial
                    28800 ; Refresh (8h)
                    7200 ; Retry (2h)
                    604800 ; Expire (7day)
                    86400 ; NegTTL (1day) (RFC2308)
                    )
                60    IN    NS    ns1.domain.tld.
                60    IN    NS    ns2.domain.tld.
                60    IN    MX    10    mx.domain.tld
                           MX    20    mx1

@           60    IN    A     1.2.4.5
mx          60    IN    A     1.2.4.5

imap        60    IN    A     1.2.4.5
smtp        60    IN    A     1.2.4.5
pop3        60    IN    A     1.2.4.5
www          IN    CNAME   h1
._imap._tcp 60    IN    SRV    5 0 143 imap.
._pop3._tcp 60    IN    SRV    5 0 110 pop3
._submission._tcp 60    IN    SRV    5 0 25 smtp
```

Файл зоны (исправленный)

```
ORIGIN domain.tld.
$TTL 3600
@ 600 IN SOA    ns1.domain.tld.    hostmaster.domain.tld. (
                    2017113001 ; Serial
                    28800 ; Refresh (8h)
                    7200 ; Retry (2h)
                    604800 ; Expire (7day)
                    86400 ; NegTTL (1day) (RFC2308)
                    )
                60    IN    NS    ns1.domain.tld.
                60    IN    NS    ns2.domain.tld.
                60    IN    MX    10    mx.domain.tld.
                           MX    20    mx1

@          60    IN    A     1.2.4.5
mx         60    IN    A     1.2.4.5
mx1        60    IN    A     1.2.4.6
imap       60    IN    A     1.2.4.5
smtp       60    IN    A     1.2.4.5
pop3       60    IN    A     1.2.4.5

_imap._tcp   60    IN    SRV    5 0 143 imap
_pop3._tcp   60    IN    SRV    5 0 110 pop3
_submission._tcp 60    IN    SRV    5 0 25 smtp
```

Ищем ошибки утилитой

```
named-checkconf zone_example_errors.txt
```

Start Of Authority

SOA - единственная запись, которая уникальна для домена (именованного пространства имен); остальные записи могут встречаться более одного раза. Эта запись описывает “точку отсчета” для домена. Содержит:

- Имя первичного авторитетного NS (nameserver);
- Почтовый адрес администратора домена - в этом месте “@” заменен на “.”, поэтому имя в адресе эл.почты лучше иметь без знаков препинания, например hostmaster;
- serial - порядковый номер версии файла. Это отправная точка для решения о синхронизации между серверами.

Важно: serial - целое число (int) в 4 байта, поэтому велика вероятность “переполнения”.

Обратное разрешение

В IP-адресе общая и частная части располагаются слева направо, а в доменном имени - наоборот.

1. IP-адрес надо развернуть: 192.168.10.1 -> 1.10.168.192. В таком виде он станет соответствовать "направлению" доменных имен.
2. централизованный домен для хранения информации об ip-адресах:
 - in-addr.arpa - для IPv4,
 - ip6.arpa - для IPv6;
3. тип записи в котором хранится имя, соответствующее адресу - PTR.

Обратная зона: 10.168.192.in-addr.arpa

Запись: 1 IN PTR www.otus.ru

Результат: 1.10.168.192.in-addr.arpa IN PTR www.otus.ru

Обратное разрешение. Ещё раз

Чтобы узнать какое имя соответствует адресу 1.2.3.4 необходимо:

- “развернуть” адрес: 1.2.3.4 -> 4.3.2.1
- сделать запрос PTR для записи 4.3.2.1.in-addr.arpa
- В ответ может быть получено ноль или более записей типа PTR, которые будут говорить какие имена указывают на этот адрес.

Репликация

В протокол DNS встроена возможность репликации зон с помощью запросов:

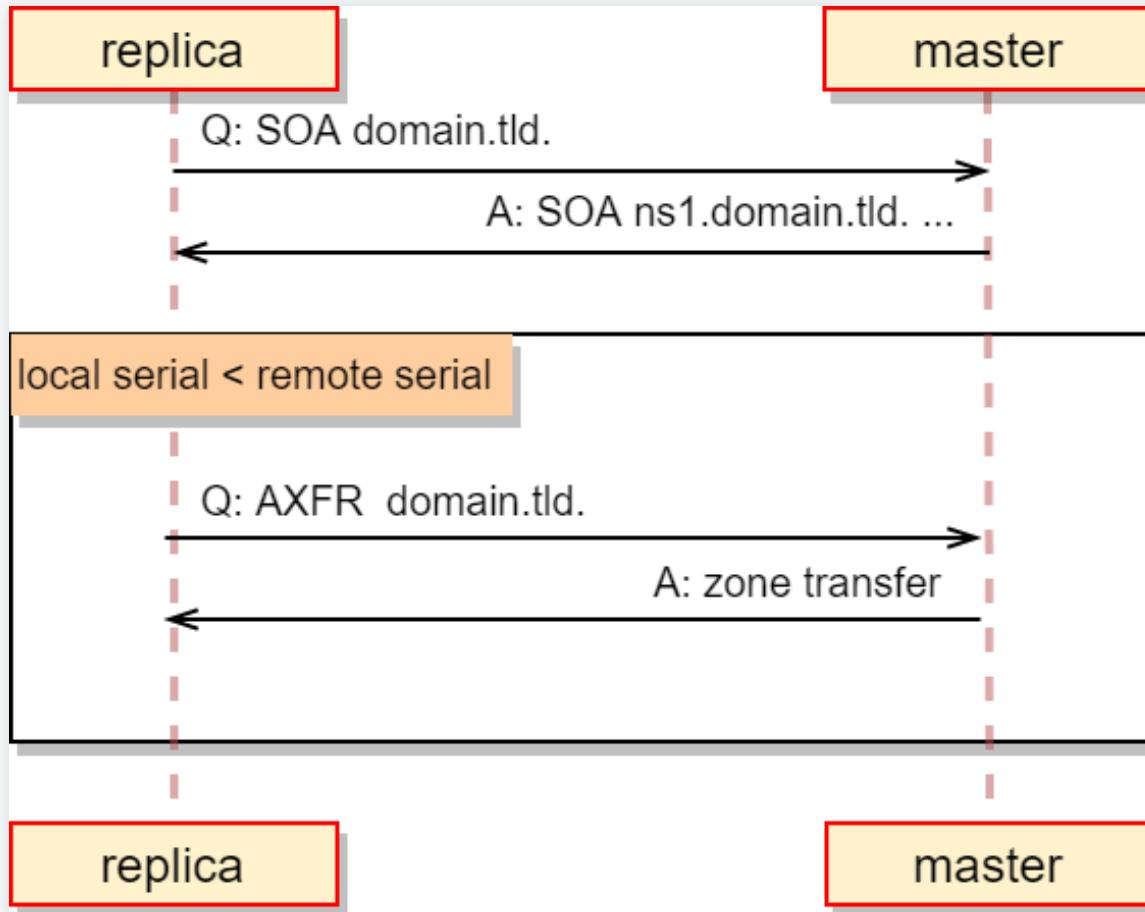
- AXFR (transfer all records);
- IXFR (incremental transfer).

Репликация происходит только при одном условии - `local_serial < remote_serial`. Проверка `serial` является частью процесса репликации. Для репликации DNS используется протокол `tcp`, т.к. важна гарантия доставки.

Репликация может быть инициирована следующими событиями:

- ручной запуск (`reload`);
- истечение `timeout` указанного в `SOA`;
- `NOTIFY`-запрос.

Репликация



Аспекты безопасности

- ограничение адресов, которым разрешены рекурсивные запросы (anti-DDoS);
- ограничение адресов, которые могут делать запросы (per zone);
- ограничение адресов, которые могут присыпать NOTIFY;
- ограничение адресов, с которых могут приходить обновления.

Утилиты

Управление:

- nsupdate
- rndc

```
// RNDC Control for client  
key "rndc-key" {
```

```
rndc flush  
rndc reload
```

Split-DNS

Иногда возникает необходимость отдавать для одной и той же зоны разные данные для одних и тех же записей. Для этого существует техника SplitDNS. В ISC bind это реализовано с помощью views.

Важно: в случае, когда определены views, не должно быть зон находящихся вне view.

Клиент может попасть (match-clients) во view основываясь на:

- адресе источника;
- адресе назначения;
- DNS TSIG-ключе.

Dynamic DNS (DDNS)

Расширение протокола DNS, которое позволяет клиентам отправлять запросы на изменение ресурсных записей (RR) первичному уполномоченному серверу непосредственно или с помощью вторичного уполномоченного сервера (предложение allow-update-forwarding в утверждении options или zone).

Утилита nsupdate позволяет сформировать пакет изменений и отослать его первичному уполномоченному серверу (имя сервера извлекается из SOA зоны).

DNSSEC (DNS Security Extensions)

Нужен для обеспечения безопасности клиентов от фальшивых DNS-данных (DNS cache poisoning, к примеру). Все ответы от DNSSEC имеют цифровую подпись. При проверке цифровой подписи DNS-клиент проверяет верность и целостность информации.

DNSSEC не шифрует данные и не обеспечивает конфиденциальность данных; только аутентификация.

Простейшая балансировка

Распределение по принципу round-robin:

- Несколько CNAME записей:

```
www1      IN      A      123.45.67.81
www2      IN      A      123.45.67.82
www      IN      CNAME    www1.example.net.
          IN      CNAME    www2.example.net.
```

- Несколько A записей:

```
www.example.net    60      IN      A      123.45.67.81
www.example.net    60      IN      A      123.45.67.82
```

Домашнее задание

- Взять стенд <https://github.com/erlong15/vagrant-bind>
- Добавить еще один клиентский сервер client2
- Завести в зоне dns.lab имена:
 - web1 указывает (смотрит) на client1;
 - web2 указывает (смотрит) на client2.
- Завести еще одну зону - newdns.lab
- завести в ней (newdns.lab) запись:
 - www, которая смотрит на обоих клиентов.
- Настроить split-dns:
 - client1 видит обе зоны, но в зоне dns.lab только web1;
 - client2 видит только dns.lab.

Дополнительно: настроить всё без выключения selinux.

Рефлексия



Отметьте 3 пункта, которые вам запомнились с вебинара



Что вы будете применять в работе из сегодняшнего вебинара?



Заполните, пожалуйста,
опрос о занятии по ссылке в чате

