DNSSEC

Введение
Принципы
Развертывание

Обзор

Что мы рассмотрим

- Проблемы, которые решает DNSSEC
- Протокол и его реализации
- Что нужно принимать во внимание при развертывании DNSSEC
- Практические проблемы связанные с развертыванием DNSSEC в реальности

Содержание

- Охват проблемы
- DNS еще раз, вкратце
- Концепции DNSSEC
- Развертывание и эксплуатация
- Проблемы (что не решено) и другие аспекты
- Сегодняшний статус DNSSEC
- Демонстрация “вживую”

Охват проблемы

Итак, в чем состоит проблема?

Отравление кэша DNS

- Ввод поддельных данных в кэш посредством:

a) возврат дополнительных (фальшивых) данных, не охваченых исходным запросом

- b) ответ кэширующему серверу с поддельными данными прежде чем тот получит ответ от авторитетного сервера

  - Первая проблем решена 20 лет назад
  - Вторая проблема теоретически очень сложна
    - До сообщения Dan Kaminsky в 2008

Охват проблемы

Каковы риски?

- Перенаправление запросов для целого домена
- Ответы о несуществующих доменах
- Захват MX
- “Удалить” большой домен (второго либо первого уровня) из кэша провайдера – DoS
- Воровство идентификации используя атаки, “обдирающие” SSL (банки, электронное правительство)
- И другие интересные вещи...

Все эти атаки были обнаружены в реальности, с ДОСТУПНЫМ кодом, их осуществляющим...
Смотрите слайды Dan Kaminsky для более полной картины

Хорошее иллюстрированное руководство
http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html

DNS: повторение

DNS, краткое повторение

- Часто используется формат файла зоны от ISC BIND – мы будем его здесь использовать

```
zone. SOA nsX.zone. hostmaster.zone.
                  ( 2009022401  ; serial
                  1d            ; refresh
                  12h           ; retry
                  1w            ; expire
                  1h )          ; neg. TTL

zone.       NS   ns.zone.
            NS   ns.otherzone.

zone.       MX   5 server.otherzone.
www.zone.    A   1.2.3.4
...
```

DNS, краткое повторение

- Структура записи:

```
NAME          [TTL]   TYPE   DATA (type specific)
-------------------------------------------------
host.zone.     3600   A      10.20.30.40
sub.zone.     86400   MX     5 server.otherzone.
```

DNS, краткое повторение

- Несколько ресурсных записей с *одинаковым именем и типом* объединяются вместе в Наборы Ресурсных Записей (RRsetы):

```
mail.zone.      MX    5 server1.zone.
mail.zone.      MX    10 server2.zone.

server1.zone.   A     10.20.30.40
server1.zone.   A     10.20.30.41
server1.zone.   A     10.20.30.42

server1.zone.   AAAA  2001:123:456::1
server1.zone.   AAAA  2001:123:456::2

server2.zone.   A     11.22.33.44
```

RRset

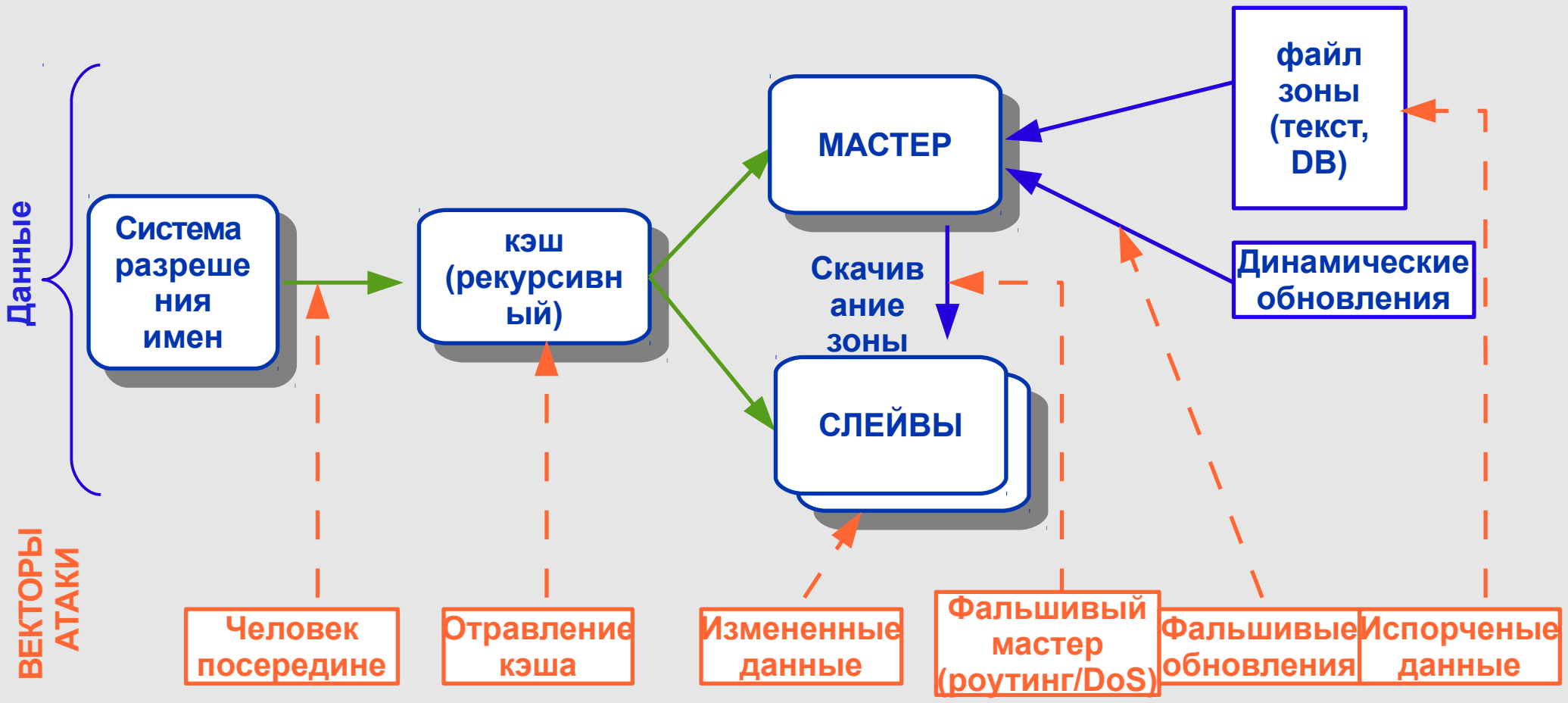
RRset

RRset

RRset

DNS – точки атаки

Поток данных DNS
Точки атаки



Концепции DNSSEC

DNSSEC в двух словах

- Поддержка аутентичности и целостности данных с помощью цифровой подписи набора ресурсных записей закрытым ключом
- Открытые ключи DNSKEY публикуются, используются для проверки подписей (RRSIG)
- Дочерние зоны подписываются их собственными закрытыми ключами
  - Аутентичность такого ключа обеспечивается *родительской* зоной, которая подписывает хэш *дочернего* ключа зоны (запись DS)
- Повторить для родительской зоны…
- Не очень сложно на бумаге
  - Но довольно сложно в эксплуатации

$$DS_{KEY} \Leftrightarrow KEY \xrightarrow{\text{подписание}} \text{данные зоны}$$

Концепции

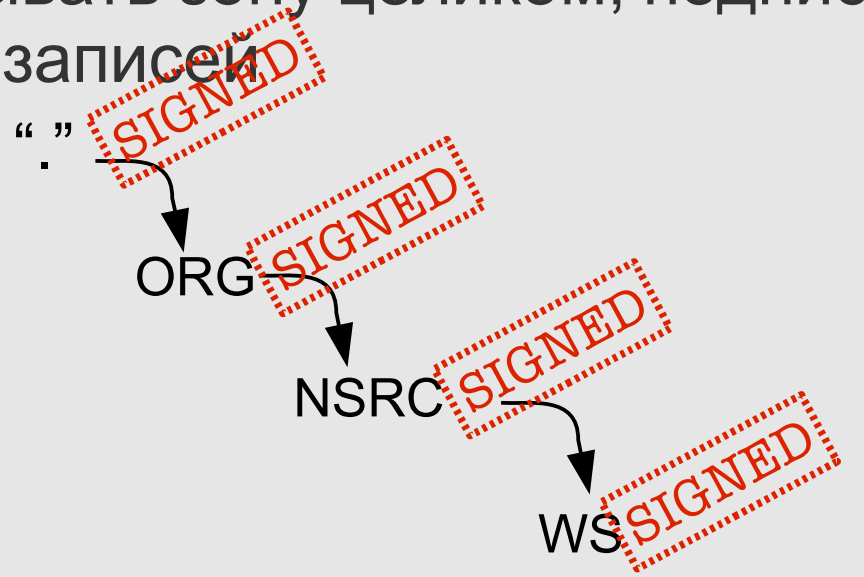
- Новые ресурсные записи (DNSKEY, RRSIG, NSEC/NSEC3 и DS)
- Новые параметры в пакетах (CD, AD, DO)
- Создание защищенной зоны
- Делегирование полномочий для подписи
- Переоформление ключей

Концепции DNSSEC

- Изменяет модель доверия DNS с “открытой” и “доверяющей” на “проверяемую”
- Использование шифрования с открытыми ключами для обеспечения:
  - Удостоверения происхождения
  - Целостности данных
  - Достоверного отрицания существования
- Нет попыток обеспечить конфиденциальность (шифрование НЕ используется)
- Обычно DNSSEC не загружает авторитетные сервера большими расчетами (это не сервера, которые *подписывают* зону)
- Отсутствуют модификации основного DNS протокола
  - Может сосуществовать с существующей инфраструктурой (EDNS0)

Концепции DNSSEC

- Построить цепочку доверия используя существующую распределенную модель DNS, основанную на делегировании
- Не подписывать зону целиком, подписывать набор ресурсных записей



- Замечание: “родитель” НЕ подписывает дочернюю зону. Родитель подписывает *указатель* (хэш) на *ключ*, используемый для подписи данных *дочерней* зоны (запись DS) record)

Новые ресурсные записи

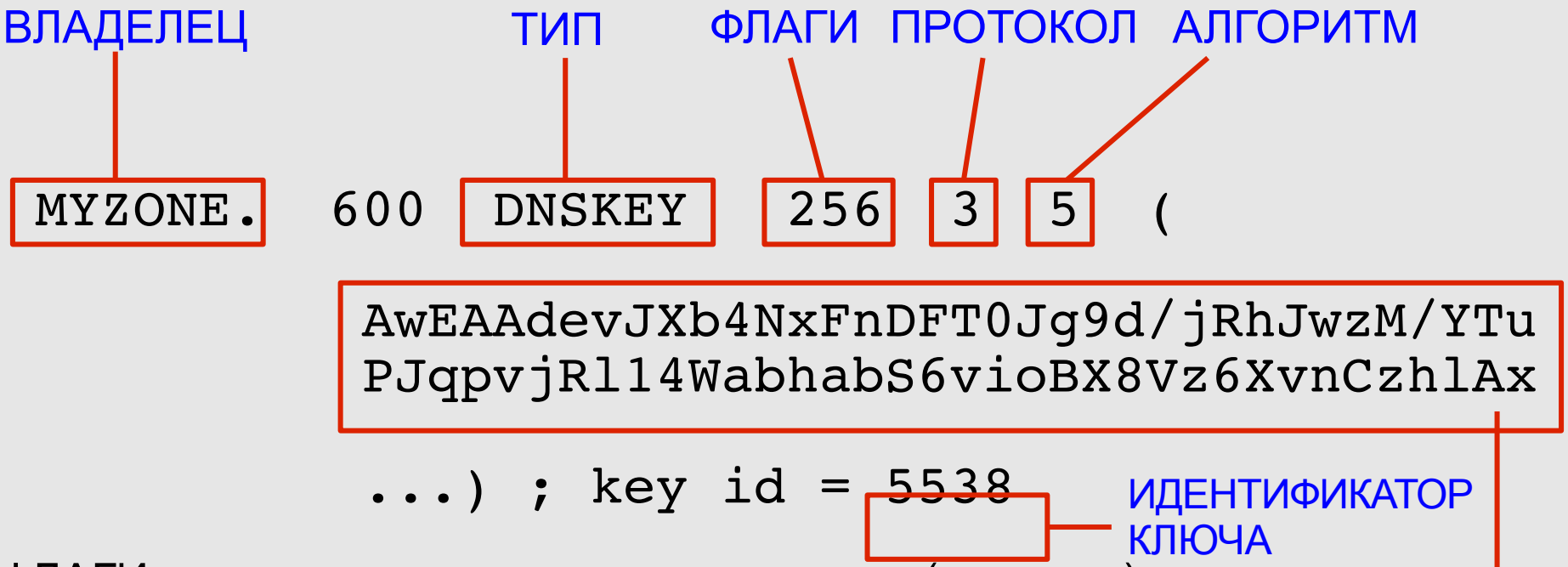
DNSSEC: новые RR

Добавляет 5 новых ресурных записей*:

1. **DNSKEY**: Открытый ключ используемый для подписывания данных зоны.
2. **RRSIG**: цифровая подпись RRseta
3. **NSEC &**
4. **NSEC3**: Возвращается как проверяемое доказательство того, что имя или типа записи не существует
5. **DS**: Делегирующий Подписант. Содержит хэш открытого ключа, используемый для подписи ключа, который в свою очередь будет использован для подписывания данных зоны. Следуйте по цепочке DS-записей RR's до тех пор пока зона “доверия” не будет достигнуто (корень дерева в идеальном случае).

*См. обсуждение Geoff Huston на http://ispcolumn.isoc.org/2006-08/dnssec.html

DNSSEC: DNSKEY RR



- ФЛАГИ определяют, как ключ используется (см. ниже)
- ПРОТОКОЛ всегда равен 3 (DNSSEC)
- АЛГОРИТМ может быть:

0 – зарезервировано

1 – RSA/MD5 (не рекомендуется)

2 – Diffie/Hellman

3 – DSA/SHA-1 (опционально)

4 – заерзервировано

5 – RSA/SHA-1 (обязательно для валидатора)

8 – RSA/SHA-256

http://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xml

DNSSEC: Два ключа, не один...

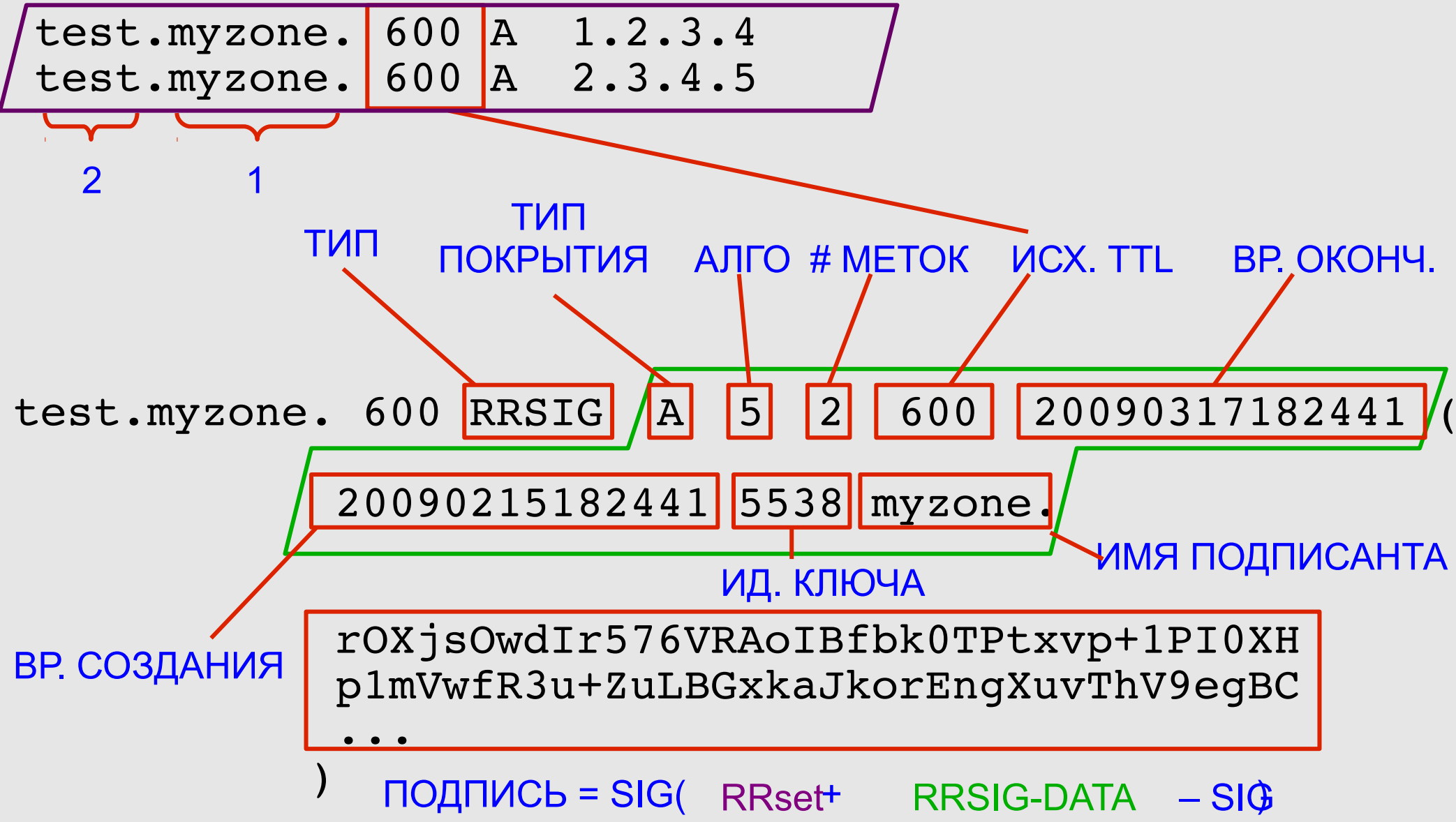
- Практически определяется по крайней мере две пары DNSKEY для каждой зоны.
- Исходно, одна пара ключей (открытый, закрытый) определялись для зоны:
  - закрытый ключ используется для подписей данных зоны (RRset-ов)
  - открытый ключ публикуется (DNSKEY) в зоне
- DNSSEC работает нормально с такой одиночной парой ключей...
- Проблема с одиночным ключом:
  - Каждый раз, когда происходит обновление ключа, запись DS, соответствующая ключу должна также быть обновлена в родительской зоне
    - Вводим Key Signing Key – ключ, подписывающий другой ключ (флаги = 257)

DNSSEC: KSK и ZSK

- Ключ, подписывающий ключ (KSK)
  - На него указывает родительская зона (при помощи DS-записи. Также называется *защищенная точка входа*
  - Используется для подписывания ключа, подписывающего зону (ZSK)
- Ключ, подписывающий зону (ZSK)
  - Подписан ключом, подписывающим ключ (KSK)
  - Используется для подписи данных зоны (RRsetов)
- Так разъединение позволяет независимо обновлять ZSK без обновления KSK, и тогда можно не вовлекать “родителя” - меньше административных взаимодействий.

$$DS_{KSK} \Leftrightarrow KSK \xrightarrow{\text{–подпись}} ZSK \xrightarrow{\text{–подпись}} RRsets$$

DNSSEC: Подпись ресурной записи - RRset подписан ZSK



DNSSEC: RRSIG

- Типичные значения (это не передовая практика, но не стандарт):
  - Время создания подписи - *1 час до “сейчас”*
  - Время окончания действия - *30 дней от “сейчас”*
  - Требуется синхронизация часов (NTP)
- Что произойдет если будет достигнуто время окончания действия подписи?
  - SERVFAIL - домен исчезает из интернета для систем разрешения имен, валидирующих DNSSEC
- Обратите внимание, *ключи **не*** имеют времени окончания действия.
- Поэтому, *регулярное* переподписывание является частью эксплуатационного процесса (а не только когда происходят изменения)
- Не все наборы ресурсных записей должны быть переподписаны в одно и то же время

DNSSEC: NSEC/NSEC3

- Доказательство несуществования при помощи NSEC и NSEC3
- Не забудьте, авторитетные серверы выдают заранее подготовленные записи. Не происходит генерирования новых записей на лету
  - NSEC указывает на следующую защищенную запись в цепочке записей
    - ➔ “не существует других записей между этой и следующей”, подпись.
  - Вся зона остортирована по алфавиту:

```
myzone.       NS    ...
ace.myzone.   A     ...
bob.myzone.   CNAME ...
cat.myzone.   A     ...
eel.myzone.   MX    ...
```

DNSSEC: NSEC/NSEC3

```
myzone. 10800 NSEC test.myzone. NS SOA RRSIG NSEC DNSKEY
myzone. 10800 RRSIG NSEC 5 1 10800 20090317182441 (
                    20090215182441 5538 myzone.

                    ZTYDLeUDMlpsp+IWV8gcUVRkIr7KmkVS5TPH
                    KPsxgXCnjnd8qk+ddXlrQerUeho4RTq8CpKV
                     ...
                    )
```

Последняя запись NSEC указывает опять на самую первую.

- Проблема:
  - Перечисление всех записей зоны (достаточно пройти по списку записей NSEC)
  - Публичный DNS не должен использоваться для хранения непубличной информации
    - Но стратегические требования разнятся

DNSSEC: NSEC/NSEC3

- Если сервер отвечает NXDOMAIN:
  - Одна из записей NSEC указывает, что имя (или расширенный шаблон имени) не существует
- Если сервер отвечает NOERROR:
  - ...и секция ответа пуста
    - ➔NSEC доказывает, что ТИП записи не существует

DNSSEC: NSEC/NSEC3

- Что там насчет NSEC3 ?
  - Мы не будет углубляться в детали:
    - Подписывается не *следующая защищенная запись*, а её **хэш**
      - Возможно доказать несуществование, но *без* раскрытия имени.
    - Это упрощенное объяснение. RFC 5155, описывающий NSEC3, весьма велик!
  - Также вводит понятие “отказа от участия” (см. раздел 6 этого RFC) для зон с большим числом делегирований
  - Нет смысла подписывать Rrsetы делегаций, о которых известно, что они не поддерживают DNSSEC.

DNSSEC: DS

- Delegation Signer (подписант делегирования)
- Хэш ключа KSK дочерней зоны
- Хранится в родительской зоне вместе с записями NS, осуществляющими делегирование дочерней зоны
- Запись DS для дочерней зоны подписывается *вместе* с остальными данными родительской зоны
  Записи NS ***НЕ*** подписываются (они просто являются указателем на дочернюю зону)

Тип дайджеста 1 = SHA-1, 2 = SHA-256

```
myzone. DS 61138 5 1
 F6CD025B3F5D030408950535 4A0115584B56D683
myzone. DS 61138 5 2
 CCBC0B557510E4256E88C01B0B1336AC4ED6FE08C826
 8CC1AA5FBF00 5DCE3210
```

дайджест = хэш(каноническое полное имя KEY RR | KEY_RR_rdata)

DNSSEC: DS

- Два хэша создаются по умолчанию:
  - 1 SHA-1 Обязательно поддерживается валидатором
  - 2 SHA-256 Обязательно поддердивается валидатором
- Новые алгоритмы тоже стандартизируются
- Это будет происходить снова и снова, по мере того как появляются доказательства ненадежности старых алгоритмов

DNSSEC: новые поля/флаги

- Обновляет протокол DNS на уровне пакетов
- Не поддерживающим DNSSEC рекурсивным серверам следует игнорировать:
  - **CD**: проверка запрещена (запрос к рекурсивному серверу не проводить валидацию, даже если подписи DNSSEC доступны и проверяемы, т.е. защищенная точка входа может быть найдена)
  - **AD**: Подлинные данные, устанавливается в ответе валидирующего сервера если ответ может быть валидирован, и клиент запросил валидацию
- Новая опция EDNS0
  - **DO**: DNSSEC OK (EDNS0 опциональный заголовок) – сообщить, что клиент поддерживает DNSSEC

Демонстрация: новые записи

Статус защищенности данных
(RFC4033 § 5 & 4035 § 4.3)

- Защищенный
  - Система разрешения имен может построить цепочку подписанных DNSKEY и DS записей начиная от доверенного защищенного якоря к набору ресурсных записей
- Не защищенный
  - Система знает, что такой цепочки не существует
- Фиктивный
  - Система полагает, что она должна суметь построить доверительную цепочку, но не может этого сделать
  - Может быть признаком взлома, но может также указывать на ошибку конфигурации либо на порчу данных
- Неопределимый
  - Нет доверенного защищенного якоря, чтобы определить защищена ли зона. Система не может определить, должны ли записи быть подписаны.

Подписание зоны...

Включение DNSSEC

- **Задействовано несколько систем**
  - Системы разрешения имен
    - Ничего не надо делать... но подробнее ниже
  - Рекурсивные кэши
    - Разрешить валидацию DNSSEC
    - Ручная конфигурация доверительных якорей (или DLV)
  - Авторитетные серверы
    - Включить поддержку DNSSEC (если нужно)
      - Подписание и ответ на запросы не обязательно делать на одной и той же машине
      - Подписывающая система может быть автономна

Подписание зоны (используя инструменты BIND)

1. Создание пар ключей
2. Включение открытых записей DNSKEY в файл зоны
3. Подписание зоны при помощи закрытого ключа ZSK
4. Опубликование зоны
5. Помещение записей DS в родительскую зону
6. Ожидание...

1. Создание ключей

```
# Создание ZSK
dnssec-keygen [-a rsasha1 -b 1024] -n ZONE  myzone

# Создание KSK
dnssec-keygen [-a rsasha1 -b 2048] -n ZONE -f KSK
 myzone
```

Эта процедура создаст 4 файла:

- Kmyzone.+005+*id_of_zsk*.key
- Kmyzone.+005+*id_of_zsk*.private
- Kmyzone.+005+*id_of_ksk*.key
- Kmyzone.+005+*id_of_ksk*.private

2. Включение ключей в зону

Включите записи DNSKEY для ZSK и KSK в зону, так что они будут подписаны вместе с остальными данными:

```
cat Kmyzone*key >>myzone
```

Или добавьте в конец файла зоны:

```
$INCLUDE “Kmyzone.+005+id_of_zsk.key”
$INCLUDE “Kmyzone.+005+id_of_ksk.key”
```

3. Подписание зоны

Подпишите вашу зону

```
# dnssec-signzone myzone
```

- dnssec-signzone выполнится со значениями по умолчанию для времени жизни подписей, серийный номер не будет увеличен по умолчанию, и закрытые ключи, используемые для подписей будут определены автоматически.
- Подписывание:
  - Отсортирует зону (по алфавиту)
  - Вставит:
    - Записи NSEC (по умолчанию)
    - Записи RRSIG (подписи для каждого RRset)
    - Записи DS из файлов ключевых записей дочерних зон (для родительской зоны используйте опцию -g
  - Создаст файл ключевых записей и файл DS записей, которые нужно послать администратору родительской зоны

3. Подписание зоны (2)

- ISC BIND

- Начиная с версии 9.7.0, автоматическое подписание зон
  - Делает жизнь намного проще
  - Создание, управление, и возобновление все еще должны выполняться отдельно

- В версии 9.8.0 появилось встроенное подписание
  - Проще интегрировать в существующую процедурную цепочку

4. Опубликование подписанной зоны

- Опубликуйте подписанную зону путем реконфигурации DNS сервера, так чтобы он подгрузил файл зоны с подписями.
- ... но вам все еще требуется отправить набор записей DS администратору родительской зоны по защищенному каналу, иначе никто не узнает, что вы используете DNSSEC

5. Помещение DS-записей в родительскую зону

- Нужно отправить защищенным образом набор DS-записей, полученных при помощи KSK, в родительскую зону
  - RFCs 4310, 5011
- ... но что делать если ваша родительская зона *не поддерживает* DNSSEC?
  - DLV

Включение DNSSEC в кэширующем сервере

- Сконфигурируйте сервер осуществлять DNSSEC-валидацию
- Протестируйте...
- Помните, валидация делается только в системе разрешения имен
- Остальные также должны разрешить DNSSEC-валидацию – не очень интересно, если это сделали только лишь вы сами!

Краткие выводы

- Создание ключей
- Подписание и опубликование зоны
- Конфигурация системы разрешения имен
- Тестирование защищенной зоны

Вопросы по предыдущему материалу?

Окончание срока действия подписи

- По умолчание, срок действия подписи равен 30 дням (BIND)
- Необходимость в регулярном переподписании:
  - Поддерживать постоянное “окно валидности” для подписей *существующих* наборов записей
  - Подписывать *новые* и *измененные* наборы записей
  - Использовать *разброс*, чтобы не требовалось переподписывать все существующие наборы записей в одно и то же время
- Ключи сами по себе НЕ имеют срока действия...
  - Но их может быть нужно возобновлять...

Возобновление ключей

- Пытайтесь минимизировать ущерб
  - Короткие сроки действия подписей
  - Регулярное возобновление ключей
- Помните: записи DNSKEY не имеют отметок времени
  - но RRSIG, подписавшая DNSKEY имеет метку времени
- Возобновление ключей вовлекает другую сторону или даже стороны:
  - Работоспособность должна поддерживаться во время возобновления
  - Дорого в эксплуатационном смысле
- RFC5011 + поддержка в BIND
- Смотрите http://www.potaroo.net/ispcol/2010-02/rollover.html

Возобновление ключей

- Два метода возобновления ключей
  - Предварительная публикация
  - Двойное подписание
- Возобновление KSK и ZSK использует разные методы (благодаря DNSSEC-Tools.org)

Возобновление ключей

- **Возобновление ZSK используя метод предварительной публикации**

1. подождать, пока старые данные в зоне не уйдут из кэшей (TTL)
2. подписать зону с KSK и опубликованным ZSK
3. подождать, пока старые данные в зоне не уйдут из кэшей
4. подкорректировать ключи в списке ключей и подписать зону с новым ZSK

Возобновление ключей

- **Возобновление KSK используя метод двойного подписания**

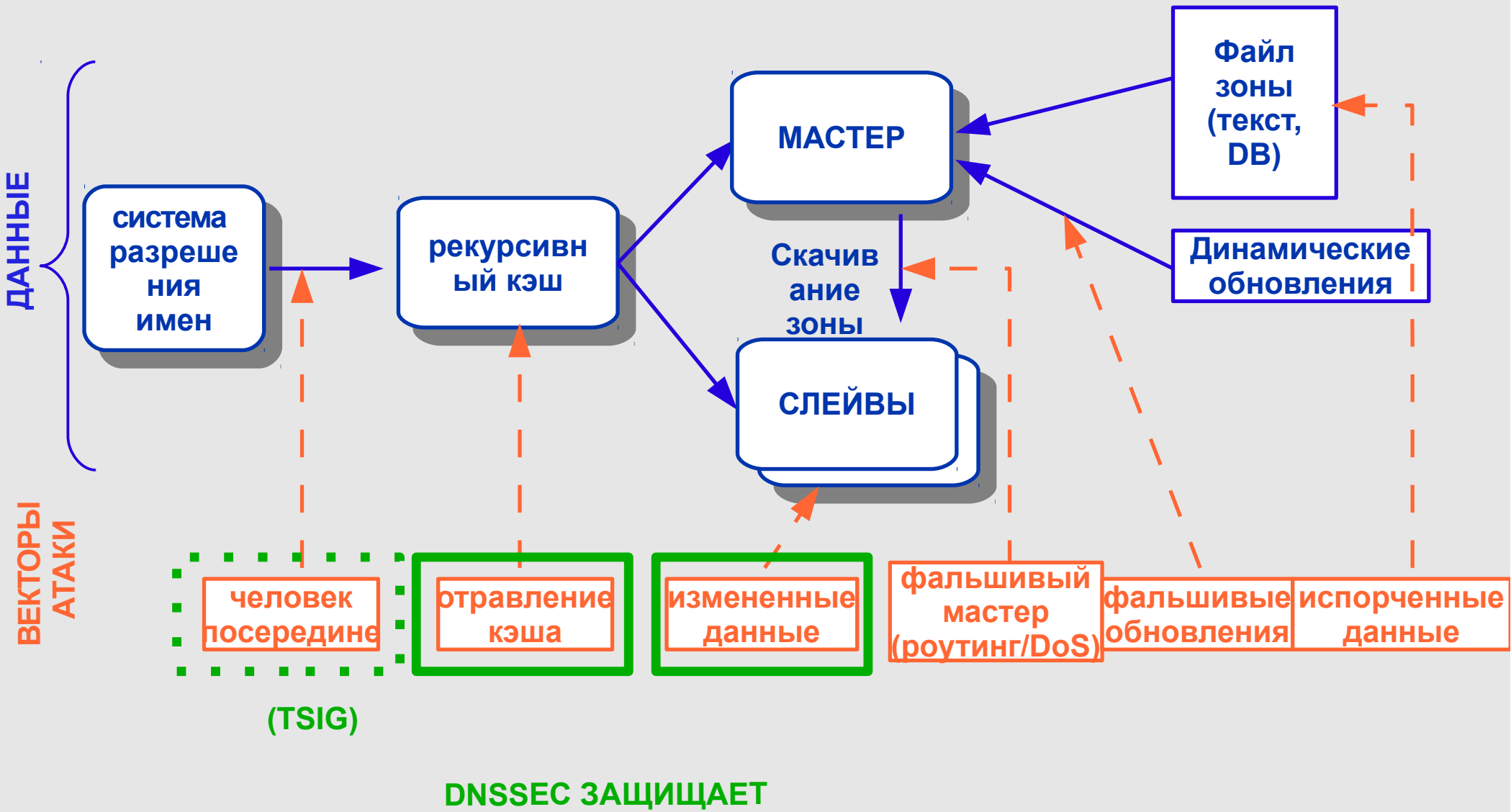
  1. подождать, пока старые данные зоны не уйдут из кэшей
  2. создать новый (опубликованный) KSK
  3. подождать, как старый набор записей DNSKEY не уйдет из кэшей
  4. перейти на новый KSK
  5. передать новый набор DS родительской зоне
  6. подождать, пока родительская зоне не опубликует новые записи DS
  7. перегрузить зону в сервере

Также возможно использовать два разных DS в родительской зоне

Автоматизированные инструменты

- К счастью, существует несколько инструментов, которые делают эксплуатацию DNSSEC достаточно гладкой
- Они не решают пока всех проблем, например взаимодействие между родительской и дочерней зонами (работа с DS, …), но они заботятся обо всех шероховатостях поддержки инфраструктуры открытых ключей
- http://www.dnssec.net/software
  - www.opendnssec.org
  - www.dnssec-tools.org
  - http://www.hznet.de/dns/zkt/
  - …

Итак, от чего защищает DNSSEC?



От чего он не защищает?

- Конфиденциальность
  - Данные не шифруются
- Сообщение между системой разрешения имен (т.е. вашей OS/рабочей станцией) и кэширующим сервером.
  - Для этого, вам нужно использовать TSIG, SIG(0), или доверять вашей системе разрешения имен
  - Он осуществляет всю валидацию от вашего имени
- Все еще нужно осуществлять валидацию самостоятельно, если вы не доверяете вышерасположенным кэширующим серверам

Валидация доверительной цепочки

Почему так долго?

Many different reasons...

- Отсутствие передовых практик. Мало опыта эксплуатации
- Опасность аварии (не подписали, не обновили), которая приведет к исчезновению зоны
- Спецификации менялись несколько раз
  - ➔NSEC допускает перечисление записей
- До 2008, DNSSEC был “решением несуществующей проблемы”
- Корневая зона была не сразу подписана (политика)
- Повышенная хрупкость – разрешение имен менее терпимо к поломкам!
- Когда валидация неуспешна, страдает клиент, а не владелец зоны

Следование доверительной цепочки (спасибо RIPE за слайд)

Locally Configured

```
Trusted Key . 8907
```

(root) .

```
.                DNSKEY (…) 5TQ3s… (8907) ; KSK
                 DNSKEY (…) lasE5… (2983) ; ZSK

                 RRSIG  DNSKEY (…)  8907 .  69Hw9…

org.             DS   7834 3 1ab15…
                 RRSIG   DS (…) . 2983
```

org.

```
org.             DNSKEY (…) q3dEw… (7834) ; KSK
                 DNSKEY (…) 5TQ3s… (5612) ; ZSK

                 RRSIG  DNSKEY (…)  7834 org.  cMas…

nsrc.org.        DS   4252 3 1ab15…
                 RRSIG  DS (…) org. 5612
```

nsrc.org.

```
nsrc.org.        DNSKEY (…) rwx002…  (4252) ; KSK
                 DNSKEY (…) sovP42…  (1111) ; ZSK

                 RRSIG  DNSKEY (…) 4252 nsrc.org.  5t...

www.nsrc.org.    A 202.12.29.5
                 RRSIG  A  (…)  1111 nsrc.org.  a3...
```

Развертывание и эксплуатация DNSSEC

Развертывание DNSSEC скучные детали

- DPS (DNSSEC: заявление о стратегии и практике)
  http://tools.ietf.org/html/rfc6841
  - Описывает дизайн, реализацию, методы и практики, определяющие эксплуатацию подписанной зоны
  - Помогает сторонним наблюдателям изучить процесс и оценить степень доверия к системе.

- Существующие эксплуатационные процедуры, которые могут быть дополнены включением поддержки DNSSEC
  - если в организации нет никаких эксплуатационных правил, шанс самому себя наказать гораздо выше

Что требуется для развертывания DNSSEC ? (2)

- Мониторинг

Препятствия к развертыванию и другие проблемы

Отсутствие опыта эксплуатации...

Все говорят о DNSSEC

- ... но совсем немногие обладают реальным практическим опытом ежедневной его эксплуатации
- Нельзя просто включить либо выключить DNSSEC
  - Простого подписания зоны уже недостаточно
  - Родительская зона должна перестать публиковать записи DS и подписи
- Варианты неполадок хорошо известны, но способы восстановления громоздки и требуют ручного вмешательства

Механизмы публикования DS

Стандартизованный способ передачи DS родительской зоне не очень распространен, иногда используют другие методы

- Загрузка с использованием SSL?
- Почта, подписанная PGP/GPG?
- EPP “расширяемый протокол предоставления информации” (RFC4310)

- Помните, передача должна быть защищенной
- Переделегирование или изменение регистранта, когда зона подписана
  - Совместное использование ключа во время перехода?
  - Запретить DNSSEC на время перехода?
  - Что если первый администратор не идет навстречу?
    - ➔ Стратегические вопросы

EDNS0 и неправильно работающие брандмауэры и DNS-серверы

DNSSEC предполагает EDNS0

- DNS-пакеты большого размера, > 512 байт
- EDNS0 не всегда распознается либо разрешается брандмауэром
- Фильтрование TCP, слишком рьяные администраторы...

- Сетевая инфраструктура во многих гостиницах не пропускает пакеты DNSSEC, или вмешивается в процесс разрешения имен

- Порталы авторизации, перенаправления

Приложения

- Приложения в большинстве своем ничего не знают о DNSSEC
  - Пользователи не могут увидеть, отчего что-то не работает
  - Вопросы в поддержку делегируются к сетевым администраторам
    - ➔ Сравните с проблемами с SSL (для не умеющих читать пользователей...)
- Существует 2 API
  - http://tools.ietf.org/id/draft-hayatnagarkar-dnsext-validator-api-07.txt
  - http://www.unbound.net/documentation/index.html

  - ➔ Расширение в Firefox, поддержка в Chrome
  - ➔ Что если приложение само устанавливает +CD ?

Защита последнего звена

- Системы разрешения имен остаются открытыми для атак типа “человек посередине”
  - Не слишком много способов обойти проблему
  - Либо доверяйте вашей системе, либо используйте TSIG, либо делайте валидацию сами
- Кое-что делается для решения этих проблем
  - Передача DNS по другим транспортным протоколам, чтобы обойти чрезмерное фильтрование
  - Проект dnssec-trigger (http://www.nlnetlabs.nl/projects/dnssec-trigger/)

OPCODE=0

?