Протоколы, сервисы и услуги
в Интернет и IP-сетях

Тема № 12
Система доменных имён. Протокол DNS

доц. каф. СС и ПД, к.т.н. С. С. Владимиров

2017 г.

Система доменных имён DNS

DNS (Domain Name System — система доменных имён)

Компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты, обслуживающих узлах для протоколов в домене (SRV-запись). Для работы протокола испольузуются протоколы TCP и UDP. Порт 53.

DNS была разработана Джоном Постелом и Полом Мокапетрисом из Института информационных наук Университета Южной Каролины в 1983 году. Оригинальное описание механизмов работы содержится в RFC 882 и RFC 883 с обновлением в RFC 973 (1986 год). В 1987 были опубликованы RFC 1034 и RFC 1035 с изменениями спецификации DNS, которые отменили RFC 882, 883 и 973 как устаревшие. Эти RFC были подписаны Полом Мокапетрисом, поэтому именно его называют автором системы DNS. Позднее в 1999 году он создал компанию Nominum, которая участвовала в разработке DNS-сервера BIND 9 (Berkeley Internet Name Domain) и протокола DHCPv6.

Распределённая база данных DNS поддерживается с помощью иерархии DNS-серверов, взаимодействующих по определённому протоколу.
Основой DNS является представление об иерархической структуре доменного имени и зонах. Каждый сервер, отвечающий за имя, может делегировать ответственность за дальнейшую часть домена другому серверу (с административной точки зрения — другой организации или человеку), что позволяет возложить ответственность за актуальность информации на серверы различных организаций (людей), отвечающих только за «свою» часть доменного имени.
Начиная с 2010 года, в систему DNS внедряются средства проверки целостности передаваемых данных, называемые DNS Security Extensions (DNSSEC). Передаваемые данные не шифруются, но их достоверность проверяется криптографическими способами. Внедряемый стандарт DANE обеспечивает передачу средствами DNS достоверной криптографической информации (сертификатов), используемых для установления безопасных и защищённых соединений транспортного и прикладного уровней.

Система доменных имён DNS

DNS важна для работы Интернета, так как для соединения с узлом необходима информация о его IP-адресе, а для людей проще запоминать буквенные (обычно осмысленные) адреса, чем последовательность цифр IP-адреса. В некоторых случаях это позволяет использовать виртуальные серверы, например, HTTP-серверы, различая их по имени запроса. Первоначально преобразование между доменными и IP-адресами производилось с использованием специального текстового файла `hosts`. Этот файл был уникальным и формировался в единственном экземпляре на сервере, размещенном в Стэнфордском исследовательском институте (SRI International). Как своего рода рудимент того времени, в усеченном виде этот файл сохраняется в большинстве операционных систем, но сейчас он зачастую содержит всего одну запись `localhost` с указанием на адрес `127.0.0.1` и иногда записи для основных multicast-адресов IPv6. Тем не менее, файл `hosts` до настоящего времени имеет преимущество перед системой DNS — если в файле `hosts` есть запись для определенного доменного имени, то обращения к системе DNS для разрешения этого имени проводиться не будет. Эта особенность широко используется администраторами при тестировании сетей, а также вирусописателями, которые встраивают в файл `hosts` запись, отправляющую пользователя на сервер злоумышленника, что позволяет ему перехватить данные пользователя, например, пароли.

Пример файла `/etc/hosts`

```
127.0.0.1       localhost
127.0.1.1       pcname

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Ключевые характеристики и понятия DNS

Ключевые характеристики DNS

- *Распределённость администрирования.* Ответственность за разные части иерархической структуры несут разные люди или организации.
- *Распределённость хранения информации.* Каждый узел сети в обязательном порядке должен хранить только те данные, которые входят в его зону ответственности, и (возможно) адреса корневых DNS-серверов.
- *Кеширование информации.* Узел может хранить некоторое количество данных не из своей зоны ответственности для уменьшения нагрузки на сеть.
- *Иерархическая структура*, в которой все узлы объединены в дерево, и каждый узел может или самостоятельно определять работу нижестоящих узлов, или делегировать (передавать) их другим узлам.
- *Резервирование.* За хранение и обслуживание своих узлов (зон) отвечают (обычно) несколько серверов, разделённые как физически, так и логически, что обеспечивает сохранность данных и продолжение работы даже в случае сбоя одного из узлов.

Домен (domain — область)

Узел в дереве имён, вместе со всеми подчинёнными ему узлами (если таковые имеются), то есть именованная ветвь или поддерево в дереве имен. Структура доменного имени отражает порядок следования узлов в иерархии; доменное имя читается слева направо от младших доменов к доменам высшего уровня (в порядке повышения значимости): вначале корневой домен (не имеющий идентификатора), ниже идут домены первого уровня (доменные зоны), затем — домены второго уровня, третьего и т. д.

Поддомен (subdomain)

Подчинённый домен. Теоретически деление может достигать глубины 127 уровней, а каждая метка может содержать до 63 символов, пока общая длина вместе с точками не достигнет 254 символов. На практике больше трёх уровней встречается крайне редко.

Ключевые понятия DNS

Ресурсная запись

Единица хранения и передачи информации в DNS. Каждая ресурсная запись имеет имя (то есть привязана к определенному Доменному имени, узлу в дереве имен), тип и поле данных, формат и содержание которого зависит от типа.

Зона

Часть дерева доменных имен (включая ресурсные записи), размещаемая как единое целое на некотором сервере доменных имен, а чаще — одновременно на нескольких серверах. Целью выделения части дерева в отдельную зону является передача ответственности за соответствующий домен другому лицу или организации (делегирование).

Делегирование

Операция передачи ответственности за часть дерева доменных имен другому лицу или организации. За счет делегирования в DNS обеспечивается распределенность администрирования и хранения. Технически делегирование выражается в выделении этой части дерева в отдельную зону, и размещении этой зоны на DNS-сервере, управляемом этим лицом или организацией.

Ключевые понятия DNS

DNS-сервер

Специализированное ПО для обслуживания DNS, а также компьютер, на котором это ПО выполняется.

DNS-клиент

Специализированная библиотека (или программа) для работы с DNS. В ряде случаев DNS-сервер выступает в роли DNS-клиента.

Авторитетность

Признак размещения зоны на DNS-сервере. Ответы DNS-сервера могут быть двух типов: авторитетные (когда сервер заявляет, что сам отвечает за зону) и неавторитетные (англ. Non-authoritative), когда сервер обрабатывает запрос, и возвращает ответ других серверов. В некоторых случаях вместо передачи запроса дальше DNS-сервер может вернуть уже известное ему (по запросам ранее) значение (режим кеширования).

DNS-запрос (DNS query)

Запрос от клиента (или сервера) серверу. Запрос может быть рекурсивным или нерекурсивным.

Система DNS содержит иерархию DNS-серверов, соответствующую иерархии зон. Каждая зона поддерживается как минимум одним авторитетным сервером DNS (authoritative — авторитетный), на котором расположена информация о домене.

Корневой домен

Корневой домен (нулевой домен, root domain)

Домен самого верхнего уровня в любой системе доменных имён. Обслуживается корневыми серверами системы доменных имен, которые располагаются в различных странах мира.

Обозначается пустым именем. При записи доменного имени, каждый домен отделяется точкой; в конце имени может присутствовать точка, которая отделяет пустое имя, соответствующее корневому домену. Если эта точка есть (например «`www.example.com.`»), то доменное имя считается полным (абсолютным). Если точки в конце имени нет («`www.example`» или «`www.example.com`»), то имя считается относительным.

Каждое интернет-приложение должно правильно обрабатывать завершающую точку, однако большинство приложений позволяют вводить доменное имя без точки в конце; обработка зависит от реализации. В простейшем случае к адресу добавляется завершающая точка, и он трактуется как абсолютный. В ряде случаев для получения полного доменного имени локальное программное обеспечение (либо приложение, либо операционная система) может присоединить к относительному имени некоторый домен по умолчанию, который определяется по доменному имени компьютера или может быть задан в настройках. Иногда в настройках может быть задано несколько таких доменов, которые перебираются по очереди, до тех пор, пока не будет найдено существующее в DNS имя. Такой подход может приводить к неоднозначности, которая может быть разрешена с помощью задания полного имени.

Корневые серверы DNS

Корневые серверы DNS

DNS-серверы, содержащие информацию о доменах верхнего уровня, указывающую на DNS-серверы, поддерживающие работу каждого из этих доменов. Основные корневые серверы DNS размещены в домене `root-servers.org` и обозначаются латинскими буквами от `A` до `M`. Они управляются различными организациями, действующими по согласованию с ICANN. Из-за существовавших в прошлом ограничений на размеры DNS-пакета (512 байт) в DNS-ответ могло быть помещено всего 13 серверов (от `A` до `M`), сейчас за этими 13 именами стоят более 200 серверов. В частности, российское зеркало сервера `F` расположено в РосНИИРОС в Москве, сервера `K` в Новосибирске, а сервера `L` в Ростове-на-Дону. Ближайший (к пользователю) адрес «зеркала» корневого сервера выбирается автоматически благодаря IP AnyCast. Так, при обращении к `K.root-servers.net`, пользователь из Новосибирска скорее всего обратится к новосибирскому серверу.

Альтернативные корневые серверы DNS

Различные организации управляют альтернативными корневыми DNS-серверами. Альтернативные системы доменных имён используют собственные DNS-серверы и управляют пространствами имён, состоящими из собственных доменов верхнего уровня. Совет по архитектуре Интернета высказался категорически против альтернативных корневых серверов в RFC 2826.

- Chaos Computer Club DNS
- OpenNIC
- Google Public DNS
- Open Root Server Confederation
- Open Root Server Network

Некоторые из альтернативных систем DNS предоставляют новые домены верхнего уровня.

Особенности доменных имен

Псевдоинтернациональные домены

Домены верхнего уровня, чьи аббревиатуры созвучны с теми или иными сокращениями. Так, домен Тувалу `.tv` широко используется как ненациональный домен верхнего уровня для телевидения; домен Федеративных Штатов Микронезии `.fm` — для FM-радиостанций; домен Острова Мэн `.im` — для интернет-мессенджеров; домен Туркменистана `.tm` — для товарных знаков; домен Молдавии `.md` — для медицинских структур; домен Западного Самоа `.ws` — для веб-сайтов; а домен Лаоса `.la` — для организаций, зарегистрированных в Лос-Анджелесе.

Зарезервированные доменные имена

Документ RFC 2606 (Reserved Top Level DNS Names — Зарезервированные имена доменов верхнего уровня) определяет названия доменов, которые следует использовать в качестве примеров (например, в документации), а также для тестирования. Кроме `example.com`, `example.org` и `example.net`, в эту группу также входят `test`, `invalid` и др.

Интернациональные доменные имена

Доменное имя может состоять только из ограниченного набора ASCII символов, позволяя набрать адрес домена независимо от языка пользователя. ICANN утвердил основанную на Punycode систему IDNA, преобразующую любую строку в кодировке Unicode в допустимый DNS набор символов.

Имя и IP-адрес не тождественны

Один IP-адрес может иметь множество имён, что позволяет поддерживать на одном компьютере множество веб-сайтов (виртуальный хостинг). Обратное тоже справедливо — одному имени может быть сопоставлено множество IP-адресов: это позволяет создавать балансировку нагрузки.

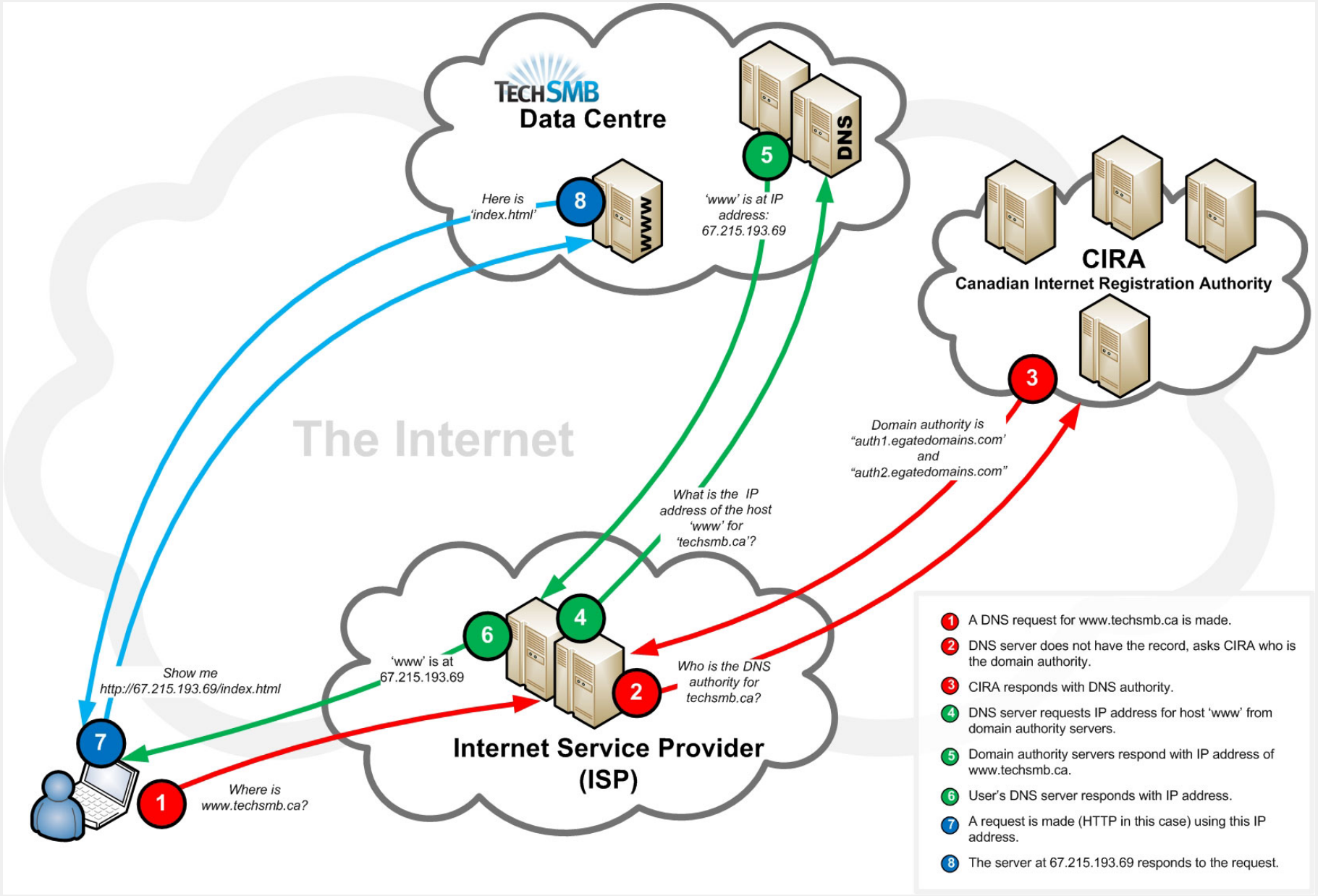
Рекурсия в DNS

Рекурсия

Алгоритм поведения DNS-сервера, при котором сервер выполняет от имени клиента полный поиск нужной информации во всей системе DNS, при необходимости обращаясь к другим DNS-серверам.

DNS-запрос может быть рекурсивным — требующим полного поиска, — и нерекурсивным (или итеративным) — не требующим полного поиска.
Аналогично, DNS-сервер может быть рекурсивным (умеющим выполнять полный поиск) и нерекурсивным (не умеющим выполнять полный поиск). Некоторые программы DNS-серверов, например, BIND, можно сконфигурировать так, чтобы запросы одних клиентов выполнялись рекурсивно, а запросы других — нерекурсивно.
При ответе на нерекурсивный запрос, а также при неумении или запрете выполнять рекурсивные запросы, DNS-сервер либо возвращает данные о зоне, за которую он ответствен, либо возвращает ошибку. Настройки нерекурсивного сервера, когда при ответе выдаются адреса серверов, которые обладают большим объёмом информации о запрошенной зоне, чем отвечающий сервер (чаще всего — адреса корневых серверов), являются некорректными и такой сервер может быть использован для организации DoS-атак.
В случае рекурсивного запроса DNS-сервер опрашивает серверы (в порядке убывания уровня зон в имени), пока не найдёт ответ или не обнаружит, что домен не существует. (На практике поиск начинается с наиболее близких к искомому DNS-серверов, если информация о них есть в кэше и не устарела, сервер может не запрашивать другие DNS-серверы.) Иногда допускается, чтобы запрошенный сервер передавал рекурсивный запрос «вышестоящему» DNS-серверу и дожидался готового ответа. При рекурсивной обработке запросов все ответы проходят через DNS-сервер, и он получает возможность кэшировать их. Повторный запрос на те же имена обычно не идет дальше кэша сервера, обращения к другим серверам не происходит вообще. Допустимое время хранения ответов в кэше приходит вместе с ответами (поле TTL ресурсной записи).
Рекурсивные запросы требуют больше ресурсов от сервера (и создают больше трафика), так что обычно принимаются от «известных» владельцу сервера узлов (например, провайдер предоставляет возможность делать рекурсивные запросы только своим клиентам, в корпоративной сети рекурсивные запросы принимаются только из локального сегмента). Нерекурсивные запросы обычно принимаются ото всех узлов сети, но содержательный ответ даётся только на запросы о зоне, размещенной на узле, а в ином случае возвращаются адреса других серверов.

Пример выполнения DNS-запроса



Записи DNS

Записи DNS, или Ресурсные записи (Resource Records, RR)

Единицы хранения и передачи информации в DNS. Каждая ресурсная запись состоит из следующих полей:

NAME	Доменное имя, к которому привязана или которому «принадлежит» данная ресурсная запись.
TTL	Time To Live — допустимое время хранения данной ресурсной записи в кэше неответственного DNS-сервера.
TYPE	Тип ресурсной записи — определяет формат и назначение данной ресурсной записи.
CLASS	Класс ресурсной записи. Теоретически считается, что DNS может использоваться не только с TCP/IP, но и с другими типами сетей, код в поле класс определяет тип сети.
RDLEN	Длина поля данных.
RDATA	Поле данных, формат и содержание которого зависит от типа записи.

Важные типы DNS-записей

A	Address record. Связывает имя хоста с адресом протокола IPv4. Например, запрос A-записи на имя `referrals.icann.org` вернёт его IPv4-адрес — `192.0.34.164`.
AAAA	IPv6 address record. Связывает имя хоста с адресом протокола IPv6. Например, запрос AAAA-записи на имя `K.root-servers.net` вернёт его IPv6-адрес — `2001:7fd::1`.
CNAME	Canonical name record. Каноническая запись имени (псевдоним). Используется для перенаправления на другое имя.
MX	Mail exchange. Указывает сервер(ы) обмена почтой для данного домена.
NS	Name server. Указывает на DNS-сервер для данного домена.
SRV	Server selection. Указывает на серверы для сервисов, используется, в частности, для Jabber и Active Directory.

Записи DNS

Важные типы DNS-записей

PTR — Pointer — запись указателя. Связывает IP-адрес хоста с его каноническим именем. Запрос в домене `in-addr.arpa` на IP-адрес хоста в обратной форме вернёт имя данного хоста. Например, для IP-адреса `192.0.34.164` запрос записи PTR `164.34.0.192.in-addr.arpa` вернёт его каноническое имя `referrals.icann.org`. В целях уменьшения объёма нежелательной корреспонденции (спама) многие серверы-получатели электронной почты могут проверять наличие PTR-записи для хоста, с которого происходит отправка. В этом случае PTR-запись для IP-адреса должна соответствовать имени отправляющего почтового сервера, которым он представляется в процессе SMTP-сессии.

SOA — Start of Authority — начальная запись зоны. Указывает, на каком сервере хранится эталонная информация о данном домене, содержит контактную информацию лица, ответственного за данную зону, тайминги (параметры времени) кеширования зонной информации и взаимодействия DNS-серверов.

Пример запроса DNS

```
user@pcname:~$ host -a sut.ru
Trying ''sut.ru''
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14859
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
   ;sut.ru.              IN      ANY
;; ANSWER SECTION:
sut.ru.      21599   IN      SOA     sut.ru. noc.sut.ru. 2022041125 10800 3600 3600000 86400
sut.ru.      21599   IN      A       91.238.228.4
sut.ru.      21599   IN      MX      10 mail.sut.ru.
sut.ru.      21599   IN      NS      ns.sut.ru.

Received 176 bytes from 8.8.8.8#53 in 12 ms
```

Обратный DNS-запрос

Обратный DNS-запрос

DNS используется в первую очередь для преобразования символьных имён в IP-адреса, но он также может выполнять обратный процесс. Для этого используются уже имеющиеся средства DNS. Дело в том, что с записью DNS могут быть сопоставлены различные данные, в том числе и какое-либо символьное имя. Существует специальный домен `in-addr.arpa`, записи в котором используются для преобразования IP-адресов в символьные имена. Например, для получения DNS-имени для адреса `11.22.33.44` можно запросить у DNS-сервера запись `44.33.22.11.in-addr.arpa`, и тот вернёт соответствующее символьное имя. Обратный порядок записи частей IP-адреса объясняется тем, что в IP-адресах старшие биты расположены в начале, а в символьных DNS-именах старшие (находящиеся ближе к корню) части расположены в конце.

Пример обратного запроса

```
user@pcname:~$ host -a 4.228.238.91.in-addr.arpa
Trying "4.228.238.91.in-addr.arpa"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22978
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;4.228.238.91.in-addr.arpa.     IN      ANY

;; ANSWER SECTION:
4.228.238.91.in-addr.arpa. 3599 IN      PTR     mail.sut.ru.

Received 68 bytes from 8.8.8.8#53 in 13 ms
```

Динамический DNS

Динамический DNS

Технология, позволяющая информации на DNS-сервере обновляться в реальном времени, и в автоматическом режиме. Она применяется для назначения постоянного доменного имени устройству (компьютеру, сетевому накопителю) с динамическим IP-адресом. Это может быть IP-адрес, полученный по DHCP или по IPCP в PPP-соединениях (например, при удалённом доступе через модем). Другие машины в Интернете могут устанавливать соединение с этой машиной по доменному имени и даже не знать, что IP-адрес изменился.
Время устаревания (TTL) для динамической записи делается очень маленьким (не более двух-трёх минут), иначе другие DNS-серверы поместят её в свой кэш, а когда она изменится — их клиенты долго будут получать устаревшую информацию.
Динамическая DNS часто применяется в локальных сетях, где клиенты получают IP-адрес по DHCP, а потом регистрируют свои имена в локальном DNS-сервере.

Протокол для обновления DNS описан в RFC 2136 и реализован, например, утилитой `nsupdate`. Для безопасной аутентификации клиента можно использовать технологию TSIG (RFC 2845), в которой используется заранее известный ключ. Минус этой технологии в том, что ключ должен быть установлен на каждом клиенте и на сервере.

Услуги динамического DNS предоставляют такие компании, как No-IP (дочерняя фирма компании Vitalwerks Internet Solutions, LLC.) и Dyn (сервис DynDNS).
Хостинговые и прочие компании, хранящие у себя DNS-информацию клиентов и позволяющие клиентам эту информацию изменять, по сути тоже предоставляют динамическую DNS. Чаще всего клиент может изменить информацию, зайдя через веб-интерфейс.

Регистратор доменных имён

Регистратор доменных имён

Организация, имеющая полномочия создавать (регистрировать) новые доменные имена и продлевать срок действия уже существующих доменных имён в домене, для которого установлена обязательная регистрация. Таковыми доменами являются:

- домен нулевого уровня (корневой домен);
- все домены верхнего уровня (первого уровня);
- некоторые домены второго уровня (например, com.ru или co.uk).

Во всех прочих доменах для создания поддоменов специальных полномочий не требуется.

Роль регистратора для корневого домена выполняет организация ICANN. Для многих доменов регистратор не единственный. При наличии нескольких регистраторов все они должны использовать единую (централизованную или распределённую) базу данных для исключения конфликтов и обеспечения уникальности доменного имени. Для того, чтобы стать регистратором доменов в зонах `.com`, `.net`, `.org`, `.biz`, `.info`, `.name`, `.mobi`, `.asia`, `.aero`, `.tel`, `.travel`, `.jobs`, необходимо получить аккредитацию ICANN.

Во многих случаях регистратор доменных имён прямо или косвенно контролируется государством. Например, регистратором домена `.mil` является подразделение Министерства обороны США (Defense Information Systems Agency), а создание российского Координационного центра национального домена курировало Министерство связи РФ.
Регистрация и продление регистрации домена осуществляется в разных доменах на разных условиях — от бесплатной до весьма дорогой (до 10 тыс. долларов). Как правило, финансовые условия одинаковые для всех владельцев (администраторов) поддоменов в рамках одного домена.

Сетевая DNS-атака: Подмена DNS-ответа; внедрение ложного DNS-сервера

Цели и угрозы

Подмена доверенных объектов сети; перехват практически любого траффика жертвы; подмена сетевых запросов/ответов.

Механизм атаки

Атакующий ждёт DNS-запроса от жертвы на хосте 1 (атакующий находится либо на хосте нарушителя 1, либо на хосте нарушителя 2; но может быть и где-либо ещё, где есть доступ к трафику хоста 1 (жертвы)).
После передачи хостом 1 DNS-запроса, атакующий принимает запрос, в котором запоминает ID и порт. Далее, такакующий отправляет ложный DNS-ответ, в котором подменяет поле IP-адрес DNS-сервера на свой IP, делая свой компьютер для жертвы валидным DNS-сервером.
Хост 1 принимает ложный DNS-ответ, принимает IP-адрес хакера за подлинный DNS-сервер и отправляет все последующие запросы ему.
Атакующий после получения DNS-запросов пересылает их на настоящий DNS-сервер, получает правильный ответ и пересылает его назад – жертве. Существует лёгкая возможность подменить в DNS-ответе IP любого запрашиваемого DNS-имени.

Сетевая DNS-атака: Подмена DNS-ответа; внедрение ложного DNS-сервера

Схема реализации атаки в случае, если доступа к трафику жертвы нет

Атакующий не дожидается DNS-запроса (он его и не получит, ибо траффик жертвы через него не проходит), а отправляет массированный поток ложных DNS-ответов, подбирая на ходу нужные порт и ID запроса (простой перебор — bruteforce). При этом в ложном ответе атакующий подменяет поле IP-адрес DNS-сервера на свой IP, делая свой компьютер для жертвы валидным DNS-сервером.

Хост 1, отправив запрос, принимает ложный DNS-ответ, в котором указан IP-адрес хакера, как IP подлинного DNS-сервера. В итоге, жертва отправляет все последующие DNS-запросы злоумышленнику.

Атакующий после получения DNS-запросов пересылает их на настоящий DNS-сервер, получает правильный ответ и пересылает его назад – жертве. Существует лёгкая возможность подменить в DNS-ответе IP любого запрашиваемого DNS-имени.

В случае, если атакующий находится за отдельным маршрутизатором и не имеет доступа к трафику клиента, но находится в том же сегменте сети, что и DNS-сервер жертвы (сама жертва), схема остаётся почти тай же. Фаза 1 заменяется на фазу массированной отправки DNS-ответов, не дожидаясь запроса жертвы. В этом случае, жертва после запроса моментально получит ответ, один из которых окажется правильным.

Сетевая DNS-атака:
Атака на кеш DNS-сервера
Подмена вышестоящего DNS-сервера (атака Каминского).

Цели и угрозы

Подмена доверенных объектов сети; перехват практически любого траффика жертвы.

Схема реализации атаки

Злоумышленник посылает на целевой DNS-сервер (жертвы) запрос, которого заведомо нет в его кеше (или поток потенциальных запросов, в один прекрасный момент случится ситуация, когда в кеше локального DNS-сервера ответа не будет).
Далее злоумышленник создает направленный шторм ложных DNS-ответов от имени одного из вышестоящих (можно взять и корневые, как на картинке) DNS-серверов. Ситуация с подменой DNS-ответа от имени корневого проще: если в предыдущей схеме нужно было подбирать порт и ID, то в текущей подбирать порт не требуется: он стандартизован и постоянен во всех соединениях между DNS-серверами: порт 53.
DNS-сервер передает DNS-запрос на вышестоящий (корневой) DNS-сервер и немедленно получает ложный DNS-ответ от атакующего.
Хост нарушителя изменяет кэш-таблицу DNS-сервера и обеспечивает прохождение трафика через подставной хост злоумышленника (адресу `top.secret.com` в кеше DNS будет соответствовать ложный IP) по тому же алгоритму, что и в описанных выше схемах.
Как видно, схема реализации атаки достаточно проста и даже не требует разворачивания собственного DNS-сервера, что делает её очень удобной и привлекательной для злоумышленников.
В результате атаки, подмена целевых DNS-ответов (IP-адресов DNS-имён) происходит не только у одного хоста-жертвы, а у всех пользователей данного DNS-сервера.

Сетевая DNS-атака:
Атака посредством отражённых DNS-запросов

Цели и угрозы

Выведение хоста жертвы (в т. ч. DNS-сервера) из строя; падение шлюза (канала доступа в Интернет)/межсетевого экрана.

Схема реализации атаки

Атака посредством отражённых DNS основывается на том, что DNS-ответ всегда в 3-4 раза длиннее запроса. В некоторых случаях (у некоторых DNS-имён) размер ответного пакета может в 10 и более раз превышать размер запроса.
Злоумышленник отправляет DNS-запросы на один или несколько сторонних DNS-серверов, которые не являются реальными объектами нападения, с подменённым IP адресом источника: IP источника равен IP хоста-жертвы.
DNS-сервера получив запрос, формируют и отправляют ответы (намного длиннее запросов) на подложный IP (т.е. жертве). Источник оказывается под мощным штормом DNS-ответов и не выдерживая нагрузки, падает.
Данный тип атаки может быть организован в условиях довольно ограниченных ресурсов, достигая 4–10-кратного эффекта усиления атаки. Если при этом злоумышленник ещё и создаст (или подберет) определенные домены, для отправки имен которых требуются DNS-пакеты огромных размеров, то отправляя запросы только на такие доменные имена, злоумышленник может достигать 100-кратного усиления эффекта атаки.
Фактически, данный тип атаки является эволюционной разновидностью обыкновенного DDoS.

Сетевая DNS-атака: Атаки типа DNS-флуд. Garbage-атаки Атаки с помощью рекурсивных DNS-запросов

Цели и угрозы

Выведение DNS-сервера — жертвы — из строя.

Основа данных видов атак — простой *DNS-флуд*: множество хостов злоумышленника посылает массированный поток запросов на целевой DNS-сервер с ложным Source IP. Стандартный компьютер способен генерировать 1000 запросов в секунду, стандартный DNS-сервер способен обрабатывать 10000 запросов в секунду. Таким образом 10 обычных домашних (или не совсем домашних) компьютеров вполне достаточно для выведения DNS-сервера из строя (условно). При том вычислить злоумышленника будет очень непросто.

Атака типа *Garbage-DNS* основывается на постоянно открытом 53 порту UDP. Схема атаки сводится к отправке злоумышленником (с множества хостов) больших (свыше 1500 байт) сетевых пакетов (не обязательно DNS). Таким образом, всё сводится к обычному DDoS, но на DNS-порт. Преимущество над обычным DDoS состоит в том, что 53 порт UDP всегда открыт, поскольку нужен для работоспособности DNS-системы.

Рекурсивная DNS-атака сводится к выявлению множества несуществующих в кеше DNS-сервера жертвы имён (возможно, фальшивых) и последующая отправка DNS-запросов с именами из этого множества. DNS-сервер в итоге вынужден пересылать подобные запросы на все соседние и вышестоящие DNS-сервера с целью получить IP заказанного хоста. В итоге, на каждый запрос сервер вынужден посылать ещё целое множество DNS-запросов другим серверам и принимать ответы от них, на что тратятся иногда в сотни раз большие ресурсы, чем отправка одного DNS-запроса. В игоге, как и в предыдущем типе атак, имея совсем не большие ресурсы, становится реальным осуществить достаточно мощную DDoS-атаку на DNS-сервер (в отличие от отражённых DNS-атак, целью которых может быть не только DNS-сервер).

Кибersквоттинг

Кибersквоттинг — захват доменов

Регистрация доменных имён, содержащих торговую марку, принадлежащую другому лицу, с целью их дальнейшей перепродажи или недобросовестного использования.

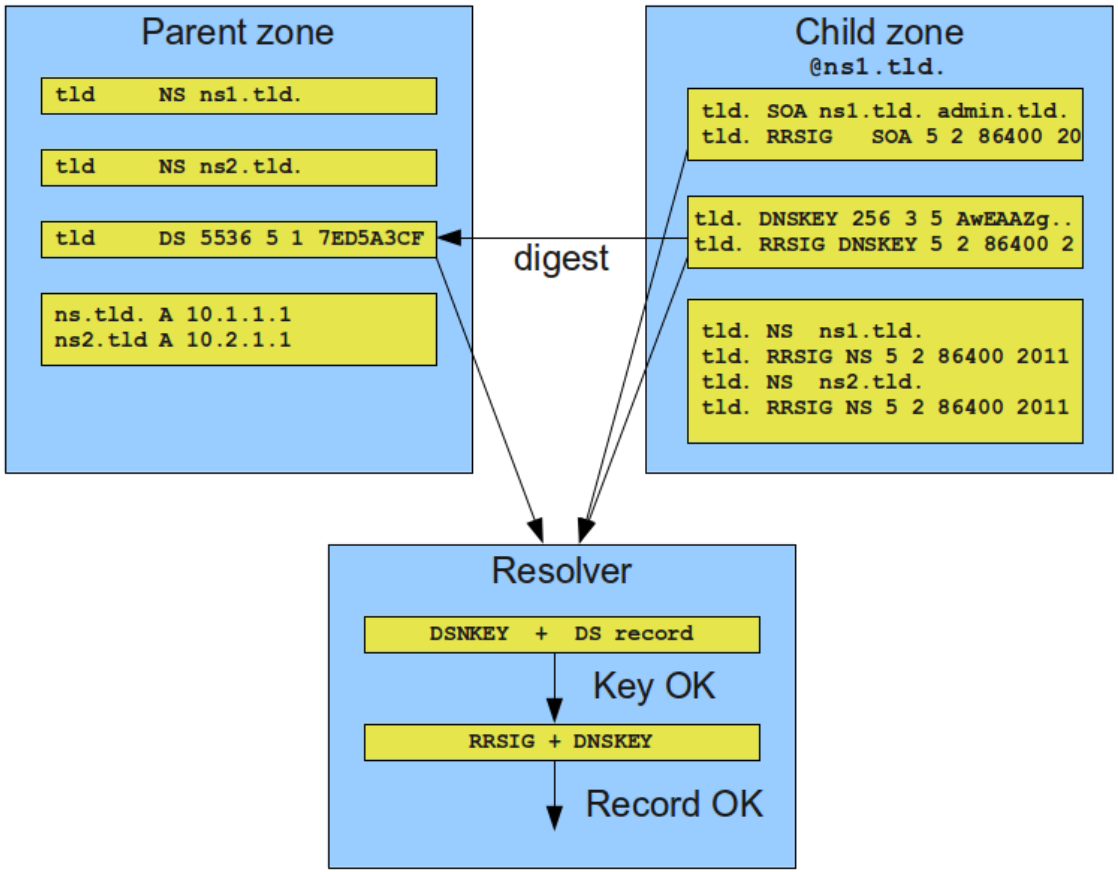
Виды кибersквоттинга

- Тайпсквоттинг — регистрация доменных имён, близких по написанию с адресами популярных сайтов в расчёте на ошибку части пользователей. Например, «`wwwsite.ru`» в расчёте на пользователя, который хотел попасть на «`www.site.ru`». При близости к очень популярным доменам тайпосквоттер может собрать на своём сайте некоторый процент «промахнувшихся» посетителей и за счёт показа рекламы заработать денег.
- Брендовый кибersквоттинг — регистрация доменных имён, содержащих товарные знаки, фирменные наименования, популярные имена собственные, то есть средства индивидуализации, охраняемые законом, а также регистрация «на перспективу», например, создатель фильма «ABC» регистрирует сайт «`ABC.com`», а кибersквоттер, с надеждой что выйдет продолжение фильма, постарается сразу зарегистрировать на себя «`ABC2.com`», «`ABC3.com`», «`ABC4.com`» и т. д. Хотя при этом у кибersквоттера есть риск лишиться домена и подвергнуться ответственности, но законные владельцы товарных знаков могут предпочесть не судиться, а выкупить захваченные домены, и цель кибersквотера будет достигнута — он заработает на этом деньги.
- Защитный кибersквоттинг — когда легальный владелец популярного сайта (товарного знака) регистрирует все доменные имена, близкие, созвучные, похожие, связанные по смыслу с его собственным доменным именем. Делается для того, чтобы не стать жертвой кибersквоттеров. Например, владелец популярного сайта «`www.firma.ru`» может захотеть также зарегистрировать домены «`firma-msk.ru`», «`firma-spb.ru`» и «`firma.org`», чтобы перенаправлять с них посетителей на свой основной сайт, а также «`anti-firma.ru`», чтобы недоброжелатели не смогли использовать его.

DNSSEC — Domain Name System Security Extensions :: https://www.dnssec.net/

Набор расширений IETF протокола DNS, позволяющих минимизировать атаки, связанные с подменой DNS-адреса при разрешении доменных имён. Направлен на предоставление DNS-клиентам аутентичных ответов на DNS-запросы (или аутентичную информацию о факте отсутствия данных) и обеспечение их целостности. При этом используется криптография с открытым ключом. Не обеспечивается доступность данных и конфиденциальность запросов.

Принцип работы



Принцип работы DNSSEC основан на использовании цифровых подписей. У администратора имеются записи о соответствии имени домена и IP-адреса. DNSSEC ставит каждой из них в строгое соответствие специальную, строго определённую последовательность символов, которая представляет собой цифровую подпись. Главная особенность цифровой подписи в том, что есть открытые, публично доступные методы проверки достоверности подписи, а вот генерирование подписи для произвольных данных требует наличия в распоряжении подписывающего секретного ключа. Поэтому проверить подпись может каждый участник системы, но подписать новые или изменённые данные на практике может только тот, у кого есть секретный ключ.

DNSSEC. Типы ключей. Ресурсные записи

Типы ключей в DNSSEC

Особенность состоит в том, что DNSSEC использует два типа ключей:

1. Ключ подписывания зоны (ZSK, Zone Signing Key).
2. Ключ для подписи набора ключей (KSK, Key Signing Key).

Сделано это, исходя из следующего: зона может быть достаточно большой чтобы удалось подобрать закрытый ключ, поэтому его надо менять чаще, и сделать его можно покороче, чтобы зоны подписывались быстрее; KSK же используется для небольших объемов данных, поэтому его можно сделать длиннее и менять реже. Тем более, что хэш открытой части KSK должен размещаться в родительской зоне, и его частые обновления нежелательны.

DNSSEC вводит концепцию подписанных зон (signed zones). Подписанная зона имеет записи общедоступного ключа DNS (DNSKEY), сигнатуру ресурсной записи (RRSIG), Next Secure (NSEC), и (опционально) Delegation Signer (подписант делегирования) (DS).

Ресурсная запись DNSKEY

Хранит публичную часть ключа ZSK. Используется для проверки записей RRSIG.

Ресурсная запись RRSIG

Содержит подпись для ресурсной записи с определенным именем, классом и типом. RRSIG определяет интервал валидности для подписи и используемого алгоритма, имя подписанта и тэг ключа, для идентификации записи DNSKEY, содержащей публичный ключ, для верификации подписи.

```
ru.    345600   IN    DNSKEY 257 3 8 AwEAAaJ6PMvWiu64aN09Y2yfZ1Y4dK...
ru.    345600   IN    RRSIG DNSKEY 8 1 345600 20181130230000 20181110230000 15506 ru. UAEdycyucsV...
```

DNSSEC. Ресурсные записи

Ресурсная запись DS

Относится к записи DNSKEY и используется в процессе аутентификации DNS DNSKEY. Хранит тэг ключа, код алгоритма и дайджест (подпись) DNSKEY RR (ключом KSK). Ресурсная запись DS и соответствующая ей DNSKEY RR имеют идентичные имена владельца, но они записываются в разных местах. Ресурсная запись DS появляется только на верхней (родительской) стороне делегирования, и является аутентификационной информацией в родительской зоне. Например, DS RR для `example.com` записывается в родительской зоне `com`, а не в самой зоне `example.com` (дочерней зон). Соответствующая DNSKEY RR записывается в зоне `example.com` (дочерняя зона). Это упрощает управление DNS-зонами, но требует специальных требований к обработке откликов для DS RR (RFC4035).

```
ru.    86400    IN    DS 15506 8 2 331CBB1932E7CF201F81AB29...
ru.    86400    IN    RRSIG DS 8 1 86400 20181124170000 20181111160000 2134 . gD1eccLYsKT1Bez42jX...
```

Ресурсная запись NSEC — Next SECure

При подписи зоны доменные имена сортируются в алфавитном порядке, к каждому из них добавляется запись NSEC, в которой указывается какое следующее доменное имя защищено и какие записи для него присутствуют в зоне. Последняя NSEC запись указывает на SOA.

Ресурсная запись NSEC3

Аналог NSEC, но доменные имена хэшируются. NSEC3 для вычисления хэша может использовать соль, помимо соли можно задать количество итераций. Увеличение количества итераций приводит к увеличению нагрузки как на резолвер, так и на авторитетный сервер, причем на последний в большей степени. Это происходит из-за того, что для возвращения NXDOMAIN, авторитетный сервер должен вычислить хэш, и не один. Процесс описан в RFC 5155.

Механизм работы DNSSEC

Хотим узнать адрес `test.bar.example.com`

1. Рекурсивно запрашиваем доменное имя у DNS-сервера (резолвера). Резолвер выставляет бит DO и запрашивает `test.bar.example.com` у корневого сервера.
2. Резолвер знает, что корневая доменная зона подписана — у него есть ее ключ или хэш ключа (т. н. trust-anchor), поэтому он запрашивает у корневого сервера DNSKEY записи для корневой зоны и сверяет их с имеющимся.
3. Корневой сервер не знает такого доменного имени, но ему известно на каких серверах располагается зона `com`. Их он и сообщает резолверу вместе с подписанной DS записью для зоны `com`.
4. Резолвер проверяет DS запись полученным и проверенным ZSK ключом корневой зоны.
5. Теперь резолвер знает, что зона `com` подписана. Он спрашивает у ее DNS сервера DNSKEY и проверяет их, после чего запрашивает `bar.example.com`. Сервер зоны `com` знает, что зона `example.com` размещена на `ns.example.com` и `ns1.example.com`. Их он возвращает резолверу вместе с DS записью.
6. Таким образом резолвер выстраивает цепочку доверия до `example.com`, где он узнает серверы имен зоны `bar.example.com` и ее DS.
7. В конце резолвер итеративно узнает адреса DNS серверов, отвечающих за `bar.example.com`, получает с них итоговую запись, проверяет всю информацию и отдает адресную запись клиенту, выставив в ответе бит AD.

При невозможности что-то проверить резолвер вернет ответ `servfail`.

1. Исходящий трафик авторитетного DNS сервера увеличивается примерно в 4 раза. Размер файла зоны после подписи вырастает в 6-7 раз.
2. Увеличение длины ключа приводит к заметному снижению qps резолвера, а записи NSEC3 аналогично влияют на авторитетный сервер.
3. DNSSec приводит к значительному увеличению DNS ответа. Сетевое оборудование должно корректно работать с DNS пакетами более 512 байт.

DNS over TLS. DNS over HTTPS

DNS поверх HTTPS (DoH)

Экспериментальный протокол для выполнения разрешения DNS по протоколу HTTPS. Целью этого метода является повышение конфиденциальности и безопасности пользователей путём предотвращения перехвата и манипулирования данными DNS с помощью MitM-атак. Google и Mozilla Foundation тестируют версии DNS по протоколу HTTPS.
В публично реализованной версии протокола Google использует HTTP GET-запросы (через HTTPS) для доступа к информации DNS с использованием кодировки DNS-запроса и параметров результата, представленных в нотации JSON.
Другая аналогичная спецификация находится в статусе интернет-проекта под эгидой IETF (RFC 8484). В этой версии протокола используются протоколы HTTP/2 и HTTPS. Записи DNS в традиционном виде пакуются в полезную нагрузку HTTPS с MIME — application/dns-message.

DNS поверх TLS (DoT)

Предлагаемый стандартный протокол для выполнения разрешения удалённой системы DNS с использованием TLS (RFC 8310). Целью этого метода является повышение конфиденциальности и безопасности пользователей путём предотвращения перехвата и манипулирования данными DNS с помощью MitM-атак.
На порт TCP:853 выполняется TLS-подключение, при этом проверка сертификата резолвера происходит с использованием системных корневых сертификатов, точно так же, как HTTPS в браузере. Это избавляет от необходимости добавлять какие-либо ключи вручную. Внутри тоннеля выполняется обычный DNS-запрос. Это создает меньше накладных расходов по сравнению с DNS over HTTPS, который добавляет HTTP-заголовки к запросу и ответу.

Источники

- Материалы с сайта `https://wikipedia.org/`
- Telecommunication technologies — телекоммуникационные технологии / Ю. А. Семенов.
  URL: `http://book.itep.ru/`
- RFC 1034. Domain Names — Concepts And Facilities.
- RFC 1035. Domain Names — Implementation And Specification.
- Пол Мокапетрис — отец DNS / Л. Черняк // Открытые системы. СУБД. №4. 2013.
  URL: `https://www.osp.ru/os/2013/04/13035567/`
- How DNS Works. URL: `http://amar-linux.blogspot.ru/2012/05/how-dns-works.html`
- DNS-атаки: полный обзор по схемам атак / А. С Лысяк // Лаборатория информационной безопасности.
  URL: `http://inforsec.ru/technical-security/network-security/77-dns-attack`
- DNSSec: Что такое и зачем. // Хабр.
  URL: `https://habr.com/post/120620/`
- Внедрение DNSSEC в информационные системы.
  URL: `https://cctld.ru/ru/domains/dnssec/`
- QUIC, TLS 1.3, DNS-over-HTTPS, далее везде. // Хабр.
  URL: `https://habr.com/company/qrator/blog/416633/`
- A cartoon intro to DNS over HTTPS / Lin Clark // Mozilla Hacks.
  URL: `https://hacks.mozilla.org/2018/05/a-cartoon-intro-to-dns-over-https/`
- Google Public DNS тихо включили поддержку DNS over TLS. // Хабр.
  URL: `https://habr.com/post/427639/`