Глава 2

Глобальная система имен

> Наконец-то у нас есть официальный список имен хостов. Теперь самое время положить конец той абсурдной ситуации, когда каждый узел сети вынужден обслуживать собственный список хостов — как правило, устаревший и отличающийся от других — для работы операционной системы или пользовательских программ.
>
> *RFC 606[1], декабрь 1973 г.*

Глобальная система доменных имен — Domain Name System, или DNS — является фундаментальным элементом Интернета. Эта система позволяет клиенту получить информацию, связанную с запрашиваемым доменным именем. Доменное имя — лучше запоминаемый, мнемонический идентификатор ресурса Сети (например, веб-сервера), в отличие от IP-адреса, записываемого в числовом виде. Наиболее распространенным запросом, обслуживаемым DNS, является получение IP-адреса устройства, связанного с именем. Поэтому функцию DNS также называют трансляцией имен в адреса.

DNS определяется набором протоколов, разработанных в IETF и опубликованных в документах RFC. С 1987 года, когда была завершена работа над основной спецификацией сегодняшней DNS (RFC 1034[2] и RFC 1035[3]), до настоящего времени

[1] RFC 606: Host Names On-line, URL: https://www.rfc-editor.org/rfc/rfc606

[2] RFC 1034: Domain Names Concepts and Facilities, URL: https://www.rfc-editor.org/rfc/rfc1034

[3] RFC 1035: Domain Names Implementation and Specification, URL: https://www.rfc-editor.org/rfc/rfc1035

было выпущено более 500 RFC, определяющих дополнительные функции системы или так или иначе связанных с ее работой. Но DNS также и глобальная распределенная база данных, хранящая сотни миллионов имен и связанных с ними ресурсов. Развиваясь вместе с самим Интернетом, DNS сегодня обслуживается более чем 16 миллионами серверов, обрабатывая несколько десятков миллионов запросов в секунду.

Нормальная работа сети Интернет немыслима без правильно функционирующей DNS. Всякий раз, когда мы набираем имя веб-сайта или отправляем электронную почту, эта система берет на себя задачу трансляции имени в цифровой адрес протокола IP, необходимый для осуществления связи между компьютерами в сети. Строго говоря, для работы IP-протокола и, соответственно, для обмена данными между компьютерами DNS не требуется. Но сегодня зависимость от этой системы настолько велика, что существенный сбой в ее работе фактически приведет к остановке самого Интернета.

Система DNS является иерархической и распределенной. Не существует единой базы данных, хранящей информацию обо всех именах, соответствующих им IP-адресах и других записях. Напротив, DNS — это миллионы баз данных, или как их чаще называют — «зон», каждая из которых содержит информацию о конкретном домене. Как правило, каждая зона обслуживается двумя или более серверами, отвечающими на запросы клиентов.

Такая архитектура DNS позволяет, во-первых, обеспечить уникальность имен, а во-вторых, распределить нагрузку и ответственность за работу системы между администраторами отдельных доменов. Каждая зона независимо обслуживается администратором зоны, отвечающим за ее содержимое, производительность и бесперебойную работу. Эта модель обеспечила долголетие системы и ее эволюционное развитие уже на протяжении более чем трех десятилетий.

Краткая история DNS

Идея использовать имена, имеющие смысловое значение, вместо числовых идентификаторов родилась почти одновременно с ARPANET. Во-первых, имя SRI-ARC или UCLA-CCn было легче запомнить, чем 002.2 или 101.65[4]. Во-вторых, в приложениях стало возможным использовать постоянные имена, не заботясь об изменениях инфраструктуры и адресов компьютеров.

Однако в то время DNS еще не существовала. Для обеспечения соответствия между именами и адресами компьютеров использовался специальный, хранимый на каждом компьютере файл, впоследствии получивший имя hosts.txt. Хотя изначально список был небольшим, насчитывавшим к концу 1973 года всего

[4] RFC 597: Host Status, URL: https://www.rfc-editor.org/rfc/rfc597

80 хостов[5], во избежание ошибок в списке требовалась координация, а именно централизация публикации и обслуживания списка. Эта функция была возложена на сетевой информационный центр в SRI, о котором мы уже говорили в первой главе. Для публикации и доступа к списку использовался протокол FTP.

К концу 1981 года, однако, число хостов перевалило за 500, и обслуживать неструктурированные имена становилось все более проблематичным. Да и список, локальная копия которого присутствовала на каждом компьютере, стал довольно громоздким.

Идея использования иерархических доменных имен родилась при решении проблемы уникальности почтовых адресов в системе доставки электронной почты в растущем Интернете. В феврале 1982 года для обсуждения возможного решения этой проблемы было организовано совещание, основные моменты которого задокументированы в RFC 805[6]. В частности, документ указывает, что «используемый в настоящее время идентификатор «user@host» должен быть расширен в «user@host.domain», где domain может представлять собой иерархию доменов».

В этом же документе можно найти наметки будущей системы DNS — систему серверов имен, в ответ на запрос возвращающих или адрес компьютера получателя почты, или же адрес сервера имен домена получателя почты (в сегодняшнем жаргоне DNS — referral, или перенаправление). Эти идеи приобрели более конкретные очертания в документе RFC 819[7], опубликованном через полгода. В частности, в документе рассматривались различные типы серверов имен, иерархия имен, а также был определен первый домен верхнего уровня — ARPA, объединявший организации, подключенные к Интернету в рамках проекта DARPA[8]. В октябре того же 1982 года появляется документ RFC 830[9] под авторством сотрудника SRI Зо-Синг Су (Zaw-Sing Su) «Распределенная система для реализации услуги имен Интернета». В нем были приведены полная кон-

---

5 Там же

6 RFC 805: Computer Mail Meeting Notes,
URL: https://www.rfc-editor.org/rfc/rfc805

7 RFC 819: The Domain Naming Convention for Internet User Applications,
URL: https://www.rfc-editor.org/rfc/rfc819

8 В 1987 г. в домене ARPA появился поддомен IN-ADDR.ARPA, который используется для обратной трансляции — трансляции адресов в доменные имена. В 2000 г. IAB опубликовал заявление, в котором предлагается использовать ARPA в качестве основного инфраструктурного домена (ранее предполагалось для этого использовать INT, но он больше подходил для международных организаций). Также предлагалось изменить аббревиатуру: ARPA теперь расшифровывалась как Address and Routing Parameters Area (область параметров маршрутизации и адресации).
URL: http://on-infrastructure-domain-and-subdomains-may-2000

9 RFC 830: A Distributed System for Internet Name Service,
URL: https://www.rfc-editor.org/rfc/rfc830

цепция и архитектура DNS, во многом оставшиеся неизменными в сегодняшней системе.

С этого момента развитие DNS происходит стремительно. Уже в ноябре 1982 года Пол Мокапетрис (Paul Mockapetris) публикует RFC 882[10] «Доменные имена — концепции и услуги» и RFC 883[11] «Доменные имена — исполнение и спецификация», которые явились первыми стандартами системы доменных имен.

В 1987-м изначальные спецификации RFC 882 и RFC 883 были дополнены с учетом опыта разработки приложений и внедрения. Обновленные стандарты, RFC 1034[12] и RFC 1035[13], и по сей день остаются основными спецификациями DNS.

Архитектура и работа DNS

Как было сказано выше, система DNS является иерархической и распределенной. Не существует единой базы данных, хранящей информацию обо всех именах и соответствующих им IP-адресах и других записях. Каждый домен, определяющий собственные имена и поддомены, является отдельной базой данных, или в терминах самой DNS — зоной. Иерархию DNS можно увидеть в доменном имени. Например, имя www.example.com. состоит из трех частей, разделенных точками. Точнее, четырех, поскольку, формально говоря, полное доменное имя всегда заканчивается точкой, обозначающей так называемый корневой домен, или корневую зону DNS. Итак:

. Корневая зона, содержащая информацию обо всех поддоменах: net, com, org, ru, su и т.д. Точнее, там содержится информация о серверах, обслуживающих эти домены.

com Домен com, содержащий информацию обо всех поддоменах, зарегистрированных в нем. В частности, о поддомене example. Опять же, эта информация включает адреса серверов, у которых можно получить дополнительную информацию о содержимом поддоменов.

---

10 RFC 882: Domain Names — Concepts and Facilities, URL: https://www.rfc-editor.org/rfc/rfc882

11 RFC 883: Domain Names — Implementation and Specification, URL: https://www.rfc-editor.org/rfc/rfc883

12 RFC 1034: Domain Names Concepts and Facilities, URL: https://www.rfc-editor.org/rfc/rfc1034

13 RFC 1035: Domain Names Implementation and Specification, URL: https://www.rfc-editor.org/rfc/rfc1035

example Домен example, содержащий информацию обо всех поддоменах, а также имена серверов, зарегистрированных непосредственно в этом домене, например www.example.com.

web Имя www-сервера и соответствующие ему IP-адреса.

Каждая зона обслуживается двумя или более серверами, отвечающими на запросы клиентов.

В DNS существует три основных типа серверов. Их роль в процессе разрешения имен показана на рис. 18.

1. Авторитетные серверы, которые обслуживают определенные зоны и дают ответы, так сказать, из первых рук. Авторитетные серверы, в свою очередь, делятся на два типа. «Мастер-сервер», его еще называют первичным, непосредственно обслуживает данные зоны. А вторичные серверы зеркалируют эти зоны для улучшения устойчивости и производительности системы. Как правило, первичный сервер обслуживается администратором зоны, а вторичные серверы — по соглашению другими операторами или компаниями, специализирующимися на оказании такого вида услуг. В ответ на запрос авторитетный сервер может либо предоставить запрашиваемую информацию (например, адрес хоста, соответствующего имени обслуживаемого домена), либо выдать отрицательный ответ, если запрашиваемое имя отсутствует в домене. Наконец, сервер может предоставить так называемый реферал (от англ. referral), или перенаправление, указав на серверы, обслуживающие поддомены, содержащиеся в имени.
2. Резолверы, также известные под именем итеративные резолверы. Эти серверы, как правило, обслуживают множество клиентов и выполняют за них основную работу по трансляции имен, а также кешируют полученные ответы для повышения производительности.
3. Резолверы-заглушки, которые выполняют простую функцию — преобразуют запрос приложения в DNS-запрос и передают его серверу, обычно итеративному резолверу, для последующей обработки. При получении ответа они делают необходимые преобразования и передают его обратно приложению. Как правило, резолверы-заглушки реализованы в виде программных библиотек или являются частью операционной системы.

Чтобы проиллюстрировать работу DNS, рассмотрим процесс преобразования имени в соответствующий ему IP-адрес. На рис. 18 приведена схема процесса[14], который происходит, когда вы набираете имя веб-сайта в окошке вашего браузера.

[14] Поэкспериментировать с различными примерами разрешения имени вам поможет утилита dig (http://ru.wikipedia.org/wiki/Dig), предоставляющая пользователю интерфейс командной строки для обращения к системе DNS. Например, процесс на рис. 18 можно увидеть с помощью команды dig + trace www.example.com (подставьте вместо www.example.com какое-либо существующее имя).

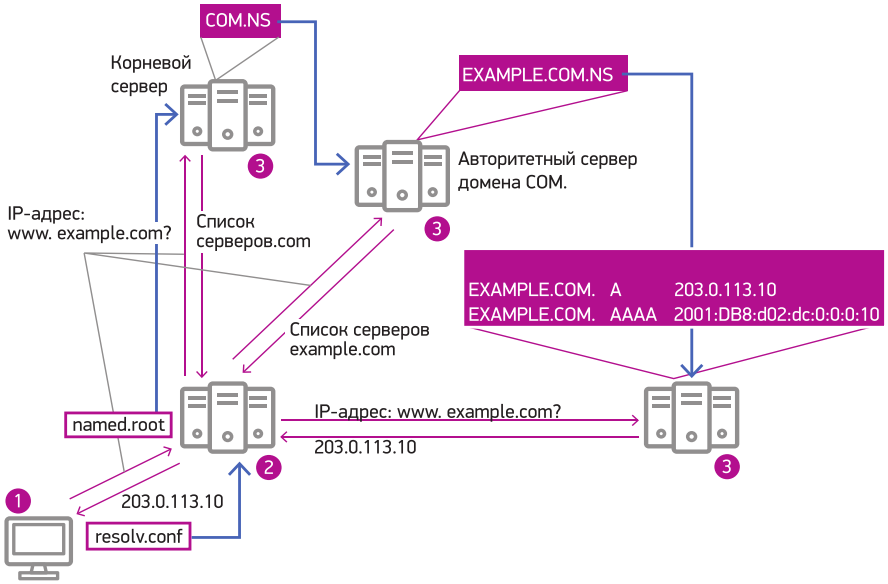

Рис. 18. Процесс трансляции имен в DNS.

Как мы уже говорили, доменные имена — это удобный способ указания ресурса. Для создания соединения TCP/HTTP нужен IP-адрес сервера, поэтому браузер должен обратиться к DNS с вопросом — каков IP-адрес сервера www.example.com (1)?

Процесс разрешения имени может быть достаточно трудоемким — для получения окончательного ответа клиенту необходимо опросить отдельные доменные базы данных, или зоны, соответствующие компонентам полного доменного имени, сужая поиск. Поэтому приложение обычно делает это не самостоятельно, а с использованием итеративного резолвера, который в ответ на полученный запрос выдает конечный результат (2).

Для получения ответа резолверу предстоит опросить все авторитетные серверы, отвечающие за соответствующие части полного доменного имени (3).

Предположим, что основная масса информации от предыдущих запросов в кеше резолвера отсутствует. В этом случае единственное, что знает резолвер в отношении полученного запроса, — адреса так называемых корневых серверов, обслуживающих корневую зону. Эти адреса содержатся в специальном файле, так называемом root hints[15]. О корневой зоне и системе корневых серверов мы

[15] В зависимости от операционной системы и программного обеспечения этот файл может иметь различные названия: named.ca, named.root, root.hints.

поговорим более подробно в разделе «Корневой уровень DNS», сейчас же лишь отметим, что данный файл содержит адреса всех 13 корневых серверов. Мастер-копия этого файла по-прежнему хранится на сайте ftp.internic.net, но резолверы автоматически поддерживают его актуальность, периодически запрашивая корневые же серверы об их текущих адресах — это так называемый первичный, или priming-запрос.

Итак, чтобы начать поиск, резолвер посылает запрос одному из корневых серверов. Эти серверы ничего не знают о существовании домена example.com и тем более адреса www.example.com. Но они сообщат, как можно связаться с серверами, обслуживающими домен следующего уровня — .com.

От одного из серверов .com резолвер узнает адреса серверов домена www.example.com, которые, в свою очередь, и ответят на запрос об IP-адресе сервера www.example.com

Хотя администратор каждой зоны действует достаточно независимо, необходима определенная координация между администраторами дочерней и родительской зон. Например, поскольку родительская зона указывает на серверы, обслуживающие дочернюю, изменения в их составе должны быть своевременно отражены в родительской зоне.

Зоны и записи

Рассмотренный нами принцип работы DNS довольно прост. Попробуем теперь более подробно изучить, что же происходит на уровне протокола.

Начнем с файла простой зоны домена www.example.com.

Таблица 1. Зона домена example.com

Имя	TTL	Класс	Тип записи	Значение
$ORIGIN example.com.				; «точка отсчета», от которой определяются все относительные имена
$TTL 86400				
@	IN	SOA	ns1.example.com.	hostmaster.example.com. (
			2001062501	; серийный номер зоны
			21600	; обновлять каждые 6 ч
			3600	; следующая попытка после 1 ч
			604800	; срок годности 1 неделя
			86400)	; минимальный TTL 1 день
	IN	NS	ns1.example.com.	; серверы имен, обслуживающие
	IN	NS	ns2.example.com.	; домен example.com

Имя	TTL	Класс	Тип записи	Значение
	IN	MX	10 mail1.example.com.	; почтовые серверы, обслуживающи
	IN	MX	20 mail2.example.com.	; домен example.com
ns1	IN	A	203.0.113.1	; IPv4 и IPv6 адреса
ns2	IN	A	203.0.113.2	; серверов имен, обслуживающих
	IN	AAAA	2001: DB8: D02: DC:0:0:0:2	; домен
server1	600 IN	A	203.0.113.10	; значение TTL уменьшено до 10 мин
server2	IN	A	203.0.113.11	; IPv4 и IPv6 адреса
		AAAA	2001: DB8: D02: DC:0:0:0: B	; серверов приложений
ftp	IN	A	203.0.113.12	
	IN	A	203.0.113.13	
mail1	IN	CNAME	server1	; псевдонимы канонического
mail2	IN	CNAME	server2	; имени server1 и server2
www	IN	CNAME	server1	
subdomain	IN	NS	ns1.example.net	; серверы имен, обслуживающие
	IN	NS	ns2.example.net	; поддомен subdomain

Каждая строка этого файла представляет собой так называемую запись ресурса (resource record). Формат записи прост:

Имя	TTL	Класс	Тип записи	Значение

Запись в столбце «Имя» определяет имя в домене зоны, например, www. При этом в нашем примере полное имя ресурса будет www.example.com. Точка в конце имени очень важна и означает полное, в отличие от относительного (относительно определения $ORIGIN) имени.

TTL — время жизни (Time To Live) записи в кеше резолвера в секундах. По прошествии этого времени запись удаляется из кеша. Значение 0 означает, что запись не должна кешироваться вообще, что, безусловно, повышает нагрузку на сервер имен, обслуживающий зону. Ведь каждый запрос клиента на разрешение этого имени будет вызывать обращение резолвера к данному авторитетному серверу.

Следующее поле — «Класс», теоретически позволяющий создавать параллельные деревья DNS. Но поскольку эта особенность практически не используется, мы не будем на ней останавливаться. В большинстве случаев на этом месте вы увидите значение IN — от Internet.

Тип записи (Record Type или RRTYPE) определяет семантику и синтаксис значения записи. Например, значение записи NS — это имя сервера имен, обслуживающего домен. Запись CNAME имеет тот же синтаксис, но ее значение является каноническим именем ресурса. В нашем примере эта запись используется для создания псевдонимов — других имен одного и того же сервера.

В таблице 2 приведены наиболее распространенные типы записей. Часть из них имеет отношение к DNSSEC, расширениям безопасности DNS, о которых мы поговорим чуть позже.

Таблица 2. Наиболее распространенные записи ресурсов DNS

Запись RR	RFC	Описание
A	RFC 1035	IPv4-адрес хоста.
AAAA	RFC 3596	IPv6-адрес хоста.
CNAME	RFC 1035	Canonical Name (каноническое имя). Позволяет определить альтернативные имена хоста (псевдонимы). Результатом ее использования является перенаправление для единственной записи.
DNAME	RFC 6672	Так же, как и CNAME, определяет перенаправление, но на уровне целой ветви DNS.
DNSKEY	RFC 4034	Определяет открытый ключ в DNSSEC.
DS	RFC 4034	Delegated Signer (делегирование подписи). Является указателем на открытый ключ дочерней зоны в DNSSEC.
MX	RFC 1035	Mail Exchanger (почтовый обмен). Определяет приоритет и имя почтового сервера, обслуживающего электронную почту для данной зоны.
NAPTR	RFC 3403	Naming Authority Pointer (указатель авторитетных имен), название не имеет ничего общего с функциональностью записи, на самом деле она используется для определения правил так называемой системы обнаружения динамического делегирования (Dynamic Delegation Discovery System, DDDS), например, при использовании протоколов VoIP или ENUM для передачи голоса по IP.
NS	RFC 1035	Name Server (сервер имен). Определяет имя авторитетного сервера имен для зоны.
NSEC	RFC 4034	Next Secure (следующая защищенная запись). Используется для подтверждения отсутствия записи в DNSSEC.
NSEC3	RFC5155	Так же, как и NSEC, используется для подтверждения отсутствия записи имени в DNSSEC, но за счет использования хешей вместо имен последующих записей предотвращает возможность получения содержимого зоны — так называемой прогулки по зоне (zone walking).
PTR	RFC 1035	Определяет имя, соответствующее IP-адресу (IPv4 или IPv6); используется в «обратном» DNS.
RRSIG	RFC 4034	Signed RRset (подписанная запись ресурса) в DNSSEC.
SOA	RFC 1035	Start of Authority (указание на авторитетность информации). Определяет имя зоны, контактный адрес электронной почты и различные параметры зоны по умолчанию: частоту обновления, «срок годности» зоны и отдельных записей.
SPF	RFC 4408	Sender Policy Framework (система политики отправителя). Определяет почтовые серверы, с которых может быть отправлена почта домена, используется для борьбы против спуфинга — маскировки и фальсификации почтового адреса-источника при посылке спама.
SRV	RFC 2872	Определяет дополнительные услуги, связанные с доменом, например, ldap, http, sip; обычно используется совместно с записью NAPTR, поддерживая обнаружение серверов этих дополнительных услуг.
TXT	RFC 1035	Текстовая информация, ассоциированная с именем, все чаще используется для расширения функциональности DNS, требуемой новыми протоколами и системами, такими как SPF или DKIM, поскольку создание и внедрение дополнительных типов записей DNS становится все более затруднительным.

Линии отреза и делегирование в DNS

Прежде чем говорить о запросах и ответах на них, кратко остановимся на том, как происходит делегирование в DNS. Как мы уже обсуждали, различные домены могут обслуживаться различными администраторами. В то же время структура DNS иерархическая, и существуют понятия родительской и дочерней зоны. Администратор родительской зоны может делегировать часть пространства имен другому администратору. Место в зоне, где происходит делегирование, называют «линией отреза» (zone cut).

В DNS эта линия может проходить только между компонентами имени. Другими словами, можно делегировать поддомен «subdomain», но нельзя делегировать все имена третьего уровня, начинающиеся с буквы s. Точнее говоря, это возможно, но для каждого имени придется задавать отдельную делегацию и, соответственно, отдельную дочернюю зону.

На первый взгляд, данная проблема носит, скорее, теоретический характер. Однако она требует практического решения в случае конфигурации обратной зоны DNS, необходимой для обслуживания так называемых обратных запросов — трансляции IP-адреса в доменное имя. Чтобы лучше понять, как происходит делегирование в DNS, давайте посмотрим на «обратный» DNS более подробно. Но сначала будет уместно сказать несколько слов о том, что такое обратная зона и обратные запросы.

Термин «обратный» используется для контраста с обычными, прямыми запросами DNS на трансляцию имени в адрес IP. При обратном запросе происходит поиск соответствующего имени для указанного IP-адреса. Зачем это может потребоваться? Например, вывод утилиты traceroute будет более «читабельным», если вместо адресов будут показаны соответствующие имена узлов, через которые проходит трафик. Обратный DNS также используется как форма аутентификации клиента — если запрос приходит от хоста с именем, это указывает на существование административного домена и его обслуживание. Данный подход используется, например, в фильтрации запросов от спамеров с компьютеров-зомби.

Однако для работы обратных запросов недостаточно конфигурации прямой зоны: единственным ключом поиска в DNS является имя, а поиск по значению (например, по IP-адресу) невозможен. Для решения этой проблемы была создана отдельная ветвь — «обратный» DNS — и специальный тип записи PTR. Ветвь эта берет свое начало в домене in-addr.arpa.

Запись PTR выглядит следующим образом:

```
$ORIGIN 113.0.203.in-addr.arpa.
...
10      IN      PTR     server1.example.com.    ; полное доменное имя сервера
```

Отправляя запрос на трансляцию имени 203.0.113.10 (не правда ли, оно похоже на доменное имя?), резолвер осуществит поиск в «обратном» DNS. Правда, сначала он преобразует адрес в «обратное» доменное имя: 10.113.0.203.in-addr.arpa. Вы, наверное, догадались, каким образом: компоненты адреса IPv4 перечислены в обратном порядке, и к полученной строке приставлен домен in-addr.arpa. Таким образом, поиск будет осуществлен сначала в корневой зоне, затем — в arpa., in-addr.arpa. и, наконец, в 113.0.203.in-addr.arpa., где и найдется запись с именем 10 и значением server1.example.com.

Все это прекрасно работало до внедрения CIDR — системы маршрутизации и соответствующей ей системы распределения адресного пространства, о которой мы говорили в предыдущей главе. Делегирование проводилось по границе октетов (каждые 8 бит адреса), что соответствовало его текстовому десятичному представлению. С появлением CIDR ситуация изменилась: стало возможным выделить клиенту сеть, скажем, размером /25. Предполагая, что мы также передаем административный контроль за обратным DNS, как же будет выглядеть обратная зона для этой сети?

Здесь и возникает проблема делегирования, которое возможно только по границе компонентов имени. То есть для 203.0.113.10 можно, например, делегировать последний компонент — 10, но это будет соответствовать сети /24!

Решение было предложено в 1998 году в RFC 2317[16], оно основано на применении псевдонимов и записей CNAME. Трюк заключается в создании промежуточного псевдодомена, например, «0–127», и его делегировании администратору сети /25. Например:

```
$ORIGIN 113.0.203.in-addr.arpa.
...
0–127   IN    NS      ns1.domainA.example.org.    ; полное доменное имя сервера
        IN    NS      ns2.example.com.            ; имен, обслуживающего зону
...
1       IN    CNAME   1.0–127.113.0.203.in-addr.arpa.
2       IN    CNAME   2.0–127.113.0.203.in-addr.arpa.
3       IN    CNAME   3.0–127.113.0.203.in-addr.arpa.
...
```

Соответственно, делегированная обратная зона для сети 203.0.113.0/25 будет выглядеть следующим образом:

[16] RFC 2317: Classless IN-ADDR.ARPA delegation, URL: https://www.rfc-editor.org/rfc/rfc2317

```
$ORIGIN 0-127.113.0.203.in-addr.arpa.
...
@       IN    NS      ns1.domainA.example.org.   ; полное доменное имя сервера
|       N     NS      ns2.example.com.
...
        IN    PTR     host1.domainA.example.org.
        IN    PTR     host2.domainA.example.org.
        IN    PTR     host3.domainA.example.org.
...
```

Возможно, вы задаете себе вопрос: как «обратный» DNS работает в случае с IPv6? Приведем пример записи PTR для такого случая. IPv6 использует собственную ветвь в ipv6.arpa, а декомпозиция адреса производится на каждые 4 бита, а не 8, как в IPv4.

```
$ORIGIN c.d.0.0.2.0.d.0.8.b.d.0.1.0.0.2.ip6.arpa.
...
b.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0   IN   PTR   server2.example.com.   ; полное доменное
                                                                              имя сервера
```

Запросы и ответы в DNS

Работа DNS происходит по схеме «запрос-ответ». Поскольку размеры запросов и ответов достаточно невелики, в основном используется транспортный протокол UDP, который, в отличие от TCP, не требует создания соединения и обеспечивает минимальные накладные расходы. Правда, ситуация постепенно меняется: размер ответов становится все больше, а с внедрением DNSSEC он может легко превысить несколько килобайт. Но об этом — позже.

И запрос, и ответ имеют одинаковую структуру, состоящую из пяти разделов: заголовок, запрос, ответ, авторитет и дополнительная информация. Эта структура представлена на рис. 19.

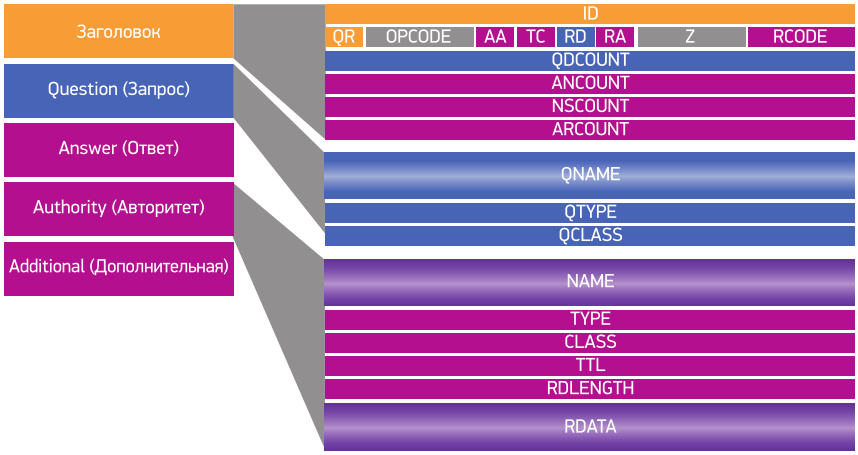

Рис. 19. Структура сообщения DNS.

Запрос использует только заголовок и первый раздел сообщения, который состоит из трех основных элементов: QNAME, QTYPE и QCLASS.

QNAME — доменное имя, именно оно в DNS всегда является поисковым ключом. Например, www.example.com.

QTYPE — тип требуемой записи, связанной с именем, указанным в QNAME. Существует также мета-тип — ANY, указывающий на запрос всех записей, связанных с именем QNAME.

QCLASS — класс: как уже говорилось, в основном это класс IN. Клиент также может указать дополнительную информацию. Например, флаг RD (Recursion Desired, «желательна рекурсия») — инструкция для запрашиваемого сервера имен, по которой следует произвести рекурсивный запрос, то есть провести полный процесс разрешения имени, показанный на рис. 18, до получения окончательного ответа. Данный флаг используется при обращении клиента (или резолвера-заглушки) к итеративному резолверу.

В зависимости от типа ответа в сообщении DNS, помимо заголовка, используются разделы ответа, авторитета и дополнительная информация. Все они имеют один и тот же формат, показанный на рис. 19.

Вы заметите, что этот формат почти в точности соответствует формату самой записи: и NAME, TYPE, CLASS, TTL и RDATA соответствуют имени, типу записи, классу, TTL и значению. RDLENGTH указывает на размер поля «Значение» записи.

Раздел ответа содержит записи, соответствующие запрашиваемому имени, если таковые найдены. В противном случае этот раздел будет пустым. Если сервер уверен, что имя не существует, то в заголовке будет установлен флаг RCODE — NXDOMAIN, указывающий, что имя не найдено. Эта ситуация отлична от случая, когда имя найдено, но не найдены запрашиваемые записи, или если имя выходит за переделы «линии отреза» и может присутствовать в дочерней зоне. В этих случаях флаг ошибки установлен не будет. Если ответ получен от авторитетного сервера для зоны, в которой происходит поиск имени, то в заголовке сообщения будет установлен флаг AA (Authoritative Answer, авторитетный ответ). Ответ, полученный из кеша, например, от итеративного резолвера, не содержит этого флага. В этом случае поле TTL будет указывать не на общий срок годности, а на оставшееся время жизни записи в кеше.

Раздел	NAME	TTL		RTYPE	RDATA
Ответ	server1.example.com.	1800	IN	A	203.0.113.11

Записи авторитетного раздела указывают на авторитетные серверы, где следует продолжить поиск для получения окончательного ответа. Такой ответ еще называют referral (реферал, или ссылка). Наиболее типичным является получение реферала от серверов доменов верхних уровней. В нашем примере (рис. 18) — это ответ от корневого сервера и от сервера домена com.

В дополнительный раздел сервер помещает информацию, которая клиенту, вероятнее всего, понадобится в будущем. Например, сервер домена .com при передаче реферала, указывающего на серверы ns1.example.com и ns2.example.com в авторитетном разделе, в дополнительном разделе укажет их IP-адреса. На запрос

Раздел	QNAME	QTYPE	QCLASS
Запрос	www.example.com.	A	IN

один из серверов домена .com (например, c.gtld-servers.net.) ответит:

Раздел	NAME	TTL		RTYPE	RDATA
Авторитетный	example.com.	86400	IN	NS	ns1.example.com.
	example.com.	86400	IN	NS	ns2.example.com.
Доп	ns1.example.com.	86400	IN	A	203.0.113.1
	ns2.example.com.	86400	IN	A	203.0.113.2

Интернационализация DNS

Система и соответствующие протоколы DNS были созданы для замены файла hosts.txt, который содержал перечень всех хостов Сети, существовавших на тот момент. Система репликации этого файла на всех компьютерах Сети не масштабировалась, а число хостов неуклонно росло. Поддержка уникальности имен, разумного времени синхронизации и обновления этих данных в глобальном масштабе — вот основные требования, предъявлявшиеся тогда к DNS. Учитывая, что все развитие Сети в то время происходило почти исключительно в англоязычной среде, требования многоязыковой поддержки не было.

Однако по мере роста и расширения Интернета ситуация начинала меняться. Требование именовать ресурсы исключительно символами ASCII казалось все более ограничительным и не соответствующим уровню глобализации Интернета.

Как сказано в одном из документов «Консорциума Юникода» — некоммерческой организации, отвечающей за разработку и сопровождение стандарта кодирования символов национальных алфавитов Юникод[17]:

[17] https://ru.wikipedia.org/wiki/Юникод

«Изначально доменные имена были ограничены набором символов ASCII. Это являлось существенным неудобством для людей, использующих другие символы. Представьте, например, что система доменных имен была бы изобретена греками и, соответственно, в URL приходилось бы использовать только греческие символы. Вместо apple.com пользователи вынужденно набирали бы что-нибудь типа αππλε.κομ. Англоговорящие пользователи должны были бы не только знать греческий алфавит, но и уметь выбирать греческие символы в соответствии с желаемым английским именем. Пришлось бы гадать, что означает то или иное имя, ведь написания не совпадают[18]».

Эта довольно абсурдная ситуация существовала до относительно недавнего времени и была актуальна для неанглоязычных пользователей Интернета. И хотя попытки интернационализировать доменные имена и предоставить возможность задавать имена на родном языке начались еще в конце 90-х годов прошлого столетия, реальная работа по стандартизации глобально применимых решений в IETF началась только в 2000 году с созданием рабочей группы по IDN (Internationalized Domain Names, интернационализированные доменные имена).

IDNA 2003

В ходе разработки стандартов рабочей группой были приняты следующие основополагающие решения:

1. Использовать Юникод как наиболее полный набор алфавитов и символов для ввода и отображения интернационализированных доменных имен. Юникод представляет собой систему кодирования символов национальных письменных языков и поддерживается «Консорциумом Юникода». Стандарт был совместно разработан с Международной организацией по стандартизации ISO, которая также поддерживает его под именем ISO/IEC 10646. Хотя оба стандарта синхронизируются, между ними есть небольшие отличия в плане требований использования и метаданных. Стандарт Юникод постоянно обновляется, в него добавляются новые языки и символы. На момент написания этой книги последней версией стандарта является Unicode 9.0[19].
2. Минимизировать изменения в архитектуре DNS, программном обеспечении элементов DNS — серверов и резолверов. Интересно отметить, что сам протокол DNS передает имена и другие поля в бинарном виде, и практически единственное ограничение, которое накладывает спецификация на имя, — это его длина. Длина имени (каждого компонента полного доменного имени) не должна превышать 63 байт, а общая длина с учетом разделителей — «.» — не должна быть больше 255 байт[20]. Широко распространенные ограничения на символы, из которых составляется доменное имя, так назы-

[18] «Unicode IDNA Compatibility Processing», http://unicode.org/reports/tr46
[19] http://www.unicode.org/versions/Unicode9.0.0
[20] RFC 2181: Clarifications to the DNS Specification, URL: https://www.rfc-editor.org/rfc/rfc2181

ваемые LDH (Letter, Digit, Hyphen — буква, цифра, дефис в формате ASCII), диктуются не самим протоколом, а приложениями, которые используют эти имена. LDH, таким образом, является «наименьшим общим знаменателем» всех ограничений. Для достижения максимальной совместимости было решено взять правило LDH в качестве требования к интернационализированному имени на уровне протокола DNS.

Таким образом, вся поддержка интернационализированных доменных имен происходит вне самой системы DNS, исключительно на уровне приложений и пользовательского интерфейса. А задача сводится к трансляции символов Юникода в формат ASCII с ограничением LDH.

Для трансляции символов Юникода в набор допустимых символов DNS был выбран алгоритм под названием Пьюникод (Punycode). Этот алгоритм определен в стандарте IETF RFC 3492[21]. Имена в Пьюникоде выглядят немного странно: например, слово «испытание» будет представлено как xn-80akhbyknj4f. Но зато они полностью соответствуют стандарту LDH. Для указания на то, что имя является ASCII-представлением (или ACE — ASCIIcompatible encoding) интернационализированного имени, используется специальный префикс xn-. Этот же алгоритм позволяет декодировать ACE-имя и получить обратно исходное слово в формате Юникод. Казалось бы, выход найден. Однако требовалось решить еще одну проблему. Дело в том, что DNS по определению осуществляет поиск до точного совпадения. Это означает, что имя в запросе (QNAME) должно полностью соответствовать имени ресурса. За исключением того, что сравнение производится без различия строчных и прописных букв.

В письменном языке все по-другому, особенно если символы кодируются в Юникоде. Отдельные символы могут выглядеть почти идентично, но иметь различные коды Юникода. Как отмечают авторы документа «Обзор и рекомендации относительно интернационализованных доменных имен»[22], символы ø (U+00F8 — LATIN SMALL LETTER O WITH STROKE) и ö (U+00F6 — LATIN SMALL LETTER O WITH DIAERESIS) считаются, например, эквивалентными в шведском языке. В то же время коды этих символов различны, соответственно, будут различны и имена с точки зрения DNS. Другой пример дополнительной сложности — некоторые символы Юникода невидимы или почти невидимы: с виду идентичные имена будут расценены системой DNS как разные. Наконец, нечувствительность DNS к регистру (другими словами, для DNS не имеет значения, строчными или прописными буквами написано имя) следует также реализовать на уровне приложений, поскольку результат трансляции Пьюникодом строчных и прописных букв будет различен. И протокол DNS, которому ничего не известно о Юникоде или Пьюникоде, будет рассматривать их как два разных имени.

---

21 RFC 3492: Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA), URL: https://www.rfc-editor.org/rfc/rfc3492

22 RFC 4690: Review and Recommendations for Internationalized Domain Names (IDNs), URL: https://www.rfc-editor.org/rfc/rfc4690

Для преодоления данных сложностей рабочей группой IDN был разработан процесс приведения в соответствие имени, набранного пользователем, включая отображение «совместимых» символов[23], приведение всех символов к строчному регистру, а также исключение некоторых символов. Этот процесс получил название Nameprep и был стандартизован в RFC 3491[24].

Группа стандартов, определяющих процесс обработки интернационализированных имен в приложениях, получила общее название IDNA 2003 — по году, когда были опубликованы составляющие ее RFC: IDNA RFC 3490[25], Nameprep RFC 3491, Punycode RFC 3492, Stringprep RFC 3454[26]. Этот процесс показан на левой стороне рис. 20.

Однако такой подход имел несколько существенных недостатков, которые становились все более очевидными по мере появления интернационализированных доменов и внедрения IDNA 2003. Как сказано в RFC 5895[27], «изначальная версия IDNA объединила и обработку на уровне пользовательского интерфейса, и собственно протокол. Она принимала любые символы, набранные пользователем, в кодировке, поддерживаемой его приложением, обеспечивала преобразование в коды Юникода, а затем без учета контекста, локальных настроек и какого-либо знания о намерениях пользователя отображала их в определенный набор других символов. И в конечном счете IDNA 2003 перекодировала эти символы Пьюникодом. Игнорирование контекста, языковых и пользовательских предпочтений в протоколе IDNA значительно облегчила жизнь разработчикам приложений. Но для потребителей и производителей доменных имен возросла вероятность некорректной трансформации исходного запроса, что явным образом нарушало так называемый принцип наименьшего сюрприза».

Пожалуй, еще более существенным недостатком IDNA 2003 явилось то, что стандарты были жестко привязаны к версии Юникода 3.2. Соответственно, при появлении новых версий (а версия 4 появилась уже в 2003 году) требовалось обновление стандартов — процесс достаточно трудоемкий.

Некоторые домены верхнего уровня (.jp, .info и несколько других) уже позволяли регистрацию интернационализированных имен, но указанные недостатки привели к тому, что в 2008 году IETF начал работу над новой серией стандартов IDNA (IDNA-bis), которая впоследствии получила имя IDNA 2008.

---

[23] См. формы нормализации Юникода: http://unicode.org/reports/tr15

[24] RFC 3491: Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN), URL: https://www.rfc-editor.org/rfc/rfc3491

[25] RFC 3490: Internationalizing Domain Names in Applications (IDNA), URL: https://www.rfc-editor.org/rfc/rfc3490

[26] RFC 3454: Preparation of Internationalized Strings («stringprep»), URL: https://www.rfc-editor.org/rfc/rfc3454

[27] RFC 5895: Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008, URL: https://www.rfc-editor.org/rfc/rfc5895

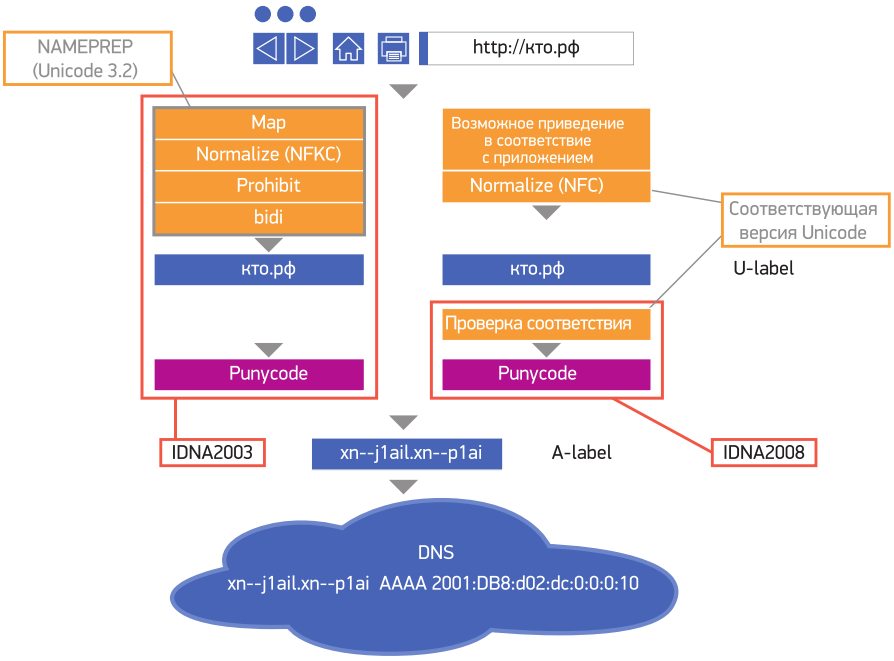

Рис. 20. Процесс обработки интернационализированных доменных имен в стандартах IDNA 2003 и IDNA 2008.

IDNA 2008

Разработчики новых стандартов выбрали несколько другой подход. Вместо того чтобы допускать практически любой символ Юникода, а затем пытаться путем отображений получить строку, готовую к трансляции в ACE, и которая, в лучшем случае, является догадкой о намерениях пользователя, в IDNA 2008 действует так называемый принцип включения. Согласно этому принципу код является недопустимым, если только он не соответствует стабильным правилам, определенным стандартам, или в редких случаях явным исключениям. К стабильным правилам относится, например, определение принадлежности символа к классу «буква или цифра». Важным здесь является то, что такие правила не зависят от используемой версии Юникода.

В результате стандарт IDNA 2008 запретил более 8000 символов, допустимых в стандарте IDNA 2003. Это ограничение репертуара символов Юникода повысило стабильность трансформации запросов, за счет чего признано обоснованным. Многие ассоциируют имена в DNS со словами, но на деле последние являются не более чем мнемониками. Соответственно, задачей IDNA является не обеспечение возможности «написать новеллу на клингонском

(или любом другом) языке[28], используя доменные имена, а поддержка создания полезных естественных мнемоник для очень широкого диапазона письменных языков»[29].

Предполагается, что приложение само, в рамках местного контекста и предпочтений пользователя, обеспечит приведение строки запроса к виду, допустимому в IDNA 2008. В качестве примеров можно перечислить кодирование строки в Юникод, если при вводе использовалась другая кодировка, приведение строки к строчным символам (IDNA 2008 включает только строчные символы), а также нормализация Юникода NFC[30].

Существенные различия между стандартами IDNA 2003 и IDNA 2008 показаны на рис. 20.

Заметим, что с использованием IDNA связан и ряд проблем. Например, IDNA открывает более широкие возможности для спуфинга (spoofing) имен, когда имя сервера внешне очень похоже на другое имя, но на самом деле использует другие символы, так называемые гомографы. Действительно, как отличишь paypal.com от paypal.com, в котором вторая буква «а» набрана кириллицей? По сравнению с обычными именами, где 1 похоже на l, а o на O, IDNA содержит гораздо больше гомографов. Решение этой проблемы требует строгого контроля со стороны регистраторов доменов, ограничения числа поддерживаемых языков в рамках домена и запрета смешивания различных языков в доменном имени.

Вопросы безопасности DNS

Итак, мы уже хорошо знаем, что система DNS является иерархической и распределенной. Оператор каждой зоны может самостоятельно определить необходимые ресурсы для обеспечения стабильности и производительности. Это делает систему в целом устойчивой и масштабируемой.

Однако специалисты в области компьютерной безопасности определяют широкий спектр угроз — потенциальных атак на систему DNS, использующих различные уязвимые места системы. Некоторые векторы атак потенциально могут нарушить функционирование глобальной DNS и, как следствие, глобального Интернета. Другие угрозы направлены больше на отдельные организации и группы пользователей.

Рассмотрим их подробнее.

---

[28] https://ru.wikipedia.org/wiki/Клингонский_язык

[29] RFC 5894: Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale, URL: https://www.rfc-editor.org/rfc/rfc5894

[30] Каноническая форма, http://unicode.org/reports/tr15

Угрозы и уязвимые места DNS

Угрозы DNS и связанные с ними атаки можно разделить на две основные категории: **атаки отказа в обслуживании** и атаки модификации данных DNS.

Цель атаки отказа в обслуживании (DoS, Denial of Service) — сделать недоступным разрешение имен для отдельных доменов. В своем типичном варианте атака генерирует громадный объем трафика, направленный на атакуемый ресурс. Это приводит к истощению ресурсов серверов домена или всей сетевой инфраструктуры, обеспечивающей его работу.

Отказ в обслуживании домена может привести к отказу обслуживания и всех дочерних доменов, связанных с ним. Также длительная недоступность первичного сервера может привести к истечению срока действия зоны на вторичных серверах и как следствие — к исчезновению зоны (и всей иерархической инфраструктуры ниже) для пользователей.

Атаки DoS, чаше всего имеющие распределенный характер, когда источники атаки расположены в различных точках Интернета (Distributed DoS, DDoS), являются атаками общего типа и могут быть направлены против любого ресурса Интернета. Интересной особенностью здесь является то, что DNS может являться как жертвой, так и средством проведения распространенного типа атаки — атаки усиления. Об этом мы поговорим чуть позже, в разделе «Противодействие атакам усиления».

Другой тип угроз — **атаки, связанные с подменой и модификацией данных DNS**. В то же время основной недостаток базового протокола DNS — слабая система защиты данных. В процессе передачи данных от сервера к клиенту они могут быть модифицированы. За счет изменения данных DNS можно создавать ложные почтовые серверы, перехватывать и перлюстрировать почтовые сообщения пользователей этого сервера. Другим примером является создание злоумышленниками веб-сайтов, имитирующих услуги электронных магазинов, банков, государственных учреждений. Последствия могут быть долгосрочными и значительными как для отдельной группы пользователей, так и для отдельного сегмента сети Интернет.

Эффективным средством борьбы с эти типом атак выступает применение расширений безопасности DNS, разработанных в рамках IETF[31] и получивших название DNSSEC.

Чтобы лучше понять значимость этой технологии для безопасности DNS, рассмотрим, насколько уязвима система в целом. Различные уязвимые места системы показаны на рис. 21, а методы защиты — на рис. 22.

[31] Internet Engineering Task Force, http://www.ietf.org

Первый вопрос — насколько защищен собственно процесс подготовки данных, редактирования и создания зоны DNS? Ошибочные данные, умышленно или случайно оказавшиеся в зоне (например, неправильные адреса веб-сайта или почтовых серверов), будут переданы пользователю в ответ на его запрос независимо от использования DNSSEC. В данном случае значение имеет уровень внутренней безопасности и защищенности процесса.

Рис. 21. Уязвимые места DNS.

Рис. 22. Методы защиты DNS.

Данные также могут быть модифицированы при передаче от мастер-сервера ко вторичным DNS-серверам, обслуживающим зону. Сегодня для проверки целостности передачи данных используется протокол TSIG (Transaction SIGnature).

Ответы от DNS-серверов также могут быть модифицированы в процессе передачи в Интернете. Но, пожалуй, самое опасное — если «неправильные» данные попадут в кеш итеративных резолверов: это называется «отравление» кеша. Дело в том, что продолжительность жизни данных в кеше, определяемая параметром TTL записи, может быть достаточно значительной. А значит, в течение всего этого времени при «отравлении» кеша пользователи будут получать подложные ответы. Более того, внедрение подложных данных в кеш резолвера не представляет особого труда, как было продемонстрировано Дэном Каминским (Dan Kaminsky) в 2008 году[32]. С тех пор в программное обеспечение большинства стандартных резолверов были введены изменения, усложняющие проведение атаки, но только DNSSEC является единственным эффективным методом защиты, поскольку позволяет криптографически проверить аутентичность данных перед загрузкой их в кеш.

Еще одна возможность публикации ложных данных в DNS — создание лжесерверов DNS. Поскольку адресация и доступ к серверам происходит по их IP-адресу, атака на систему маршрутизации, а именно «захват» соответствующего адресного префикса может привести к перенаправлению части запросов DNS к DNS-серверу злоумышленника. Соответственно, злоумышленник получает возможность публикации ложных данных с далеко идущими последствиями. Такая угроза особенно значительна при масштабном внедрении технологии anycast (аникаст), когда несколько серверов, расположенных в различных частях Интернета, используют один и тот же IP-адрес. В этом случае отличить достоверный DNS-сервер от лжесервера становится сложнее.

Наконец, ответы резолвера на запросы клиентов могут быть также модифицированы. DNSSEC здесь не поможет, если пользователь не производит проверку подлинности ответов самостоятельно, а полагается на резолвер. Правда, канал между резолвером и пользователем, как правило, находится под административным контролем сервис-провайдера или администратора корпоративной сети и зачастую имеет высокую степень защиты, например с помощью VPN.

Повышение устойчивости системы и противодействие атакам усиления

Учитывая критичность системы DNS для нормальной работы Интернета, ее устойчивость имеет большое значение. Различные требования (см., например, Технические требования к авторитетным серверам имен[33] или

[32] https://www.cnet.com/news/privacy/researcher-offers-insight-into-dns-flaw/

[33] Technical requirements for authoritative name servers, https://www.iana.org/help/nameserver-requirements

RFC 1912[34]) устанавливают, что любая зона обслуживается как минимум двумя серверами, которые находятся в разных сетях. Тем самым гарантируется работоспособность в случае отказа одного из компонентов. Также рекомендуется использовать различное программное обеспечение для минимизации ситуаций, когда ошибка или использованная уязвимость ПО приводит к выходу из строя всей системы. Однако отказ или недоступность одного из серверов (например, вследствие потери связности или поломки) являются только одной из проблем. Другой фактор, который может повлиять на устойчивость и производительность системы, — атаки «отказа в обслуживании» DoS. Чтобы эффективно им противостоять, необходимо иметь как можно более распределенную систему серверов.

Также при оценке производительности и устойчивости системы нужно принимать во внимание такие факторы, как устойчивость к перегрузкам и время реакции — промежуток времени, необходимый для получения ответа клиентами, учитывая их географическую распределенность.

Мы коснулись атак DDoS, когда обсуждали угрозы и уязвимые места DNS. Тогда же было отмечено, что в плане атак усиления, которые являются специальным типом атак отказа в обслуживании DDoS, DNS играет двоякую роль.

Во-первых, атаки усиления зачастую выбирают DNS в качестве цели. Атака такого рода может привести к отказу обслуживания не только атакуемого домена, но и всех дочерних доменов, связанных с ним. Это, в свою очередь, сделает практически недоступными информационные ресурсы (например, веб и электронную почту), связанные с соответствующими именами. Во-вторых, DNS сам является ключевым элементом так называемых рефлекторных атак, играя роль усилителя.

Кратко остановимся на самих атаках.

Рефлекторные атаки с усилением

Суть рефлекторной атаки достаточно проста и базируется на трех основных ингредиентах:

1. **Использование возможности «спуфинга»** — подмены IP-адреса отправителя на адрес «жертвы» с протоколами UDP или ICMP, обеспечивающими передачу дейтаграмм без создания соединения. Такие широко распространенные услуги Интернета, как SNMP и DNS, используют именно эти протоколы передачи. Ключевым фактором здесь является отсутствие необходимости «рукопожатия», как, например, в случае с TCP, для начала передачи данных.
2. **Рефлекторы и усилители.** Поскольку режимом работы многих услуг, основанных на протоколе UDP, является «запрос-ответ», при подмене адреса отправителя на адрес «жертвы» ответ на запрос будет доставлен именно

[34] RFC 1912: Common DNS Operational and Configuration Errors, URL: https://www.rfc-editor.org/rfc/rfc1912

туда. Представьте, что такого типа запросы посланы с различных точек Интернета. Все ответы на эти запросы будут направлены на один адрес, обеспечивая значительную концентрацию трафика в направлении «жертвы». «Хороший» рефлектор также является усилителем, когда размер ответа во много раз превышает размер запроса (см. таблицу 2). Это позволяет создать асимметрию, когда относительно незначительный трафик запросов превращается в мощный ответный поток.

3. **Ботнеты.** Для эффективных атак такого рода необходима хорошо распределенная сеть источников. Инфицированные компьютеры, объединенные в ботнеты, являются для этого прекрасной стартовой площадкой.

Смешав эти ингредиенты, мы получим рефлекторную атаку с усилением. Работает она следующим образом.

Таблица 3. Типичные усилители

Протокол	Протокол/Порт	Наблюдаемый коэффициент усиления	Ориентировочное число усилителей в Интернете
DNS	UDP/53	100	7–15 млн
SNMPv2	UDP/161	12	5 млн
NTP	UDP/123	4600	1,5 млн
CHARGEN	UDP/19	360	90 тыс.
SSDP	UDP/1900	80	3,7 млн

Источник: C. Rossow, «Amplification Hell: Revisiting Network Protocols for DDoS Abuse» (https://www.christian-rossow.de/publications/amplification-ndss2014.pdf)

- Выбирается один или несколько усилителей-рефлекторов. В качестве таковых могут служить DNS-серверы и «открытые» резолверы (о них мы поговорим чуть позже), готовые предоставить ответ на запрос со значительным коэффициентом усиления. Типичным является усиление трафика в 30–60 раз.
- Компьютеры ботнета получают инструкцию начать атаку на жертву. По команде они посылают заданные запросы выбранным усилителям-рефлекторам, подменяя адрес отправителя на IP-адрес жертвы.
- Серверы-рефлекторы отвечают на невинные с виду запросы, которые реально поступают от различных клиентов, в то же время обрушивая усиленный трафик на жертву. Поскольку обычно используется большое количество рефлекторов, расположенных в различных точках Интернета, объем трафика увеличивается по мере приближения к жертве.
- Объем сгенерированного трафика превышает пропускную способность каналов или максимальную производительность атакуемого сервера, тем са-

мым вызывая отказ в обслуживании или невозможность предоставления услуг. Услуга, подвергшаяся атаке, на какое-то время перестает быть доступной в Интернете. Этот процесс схематически представлен на рис. 23.

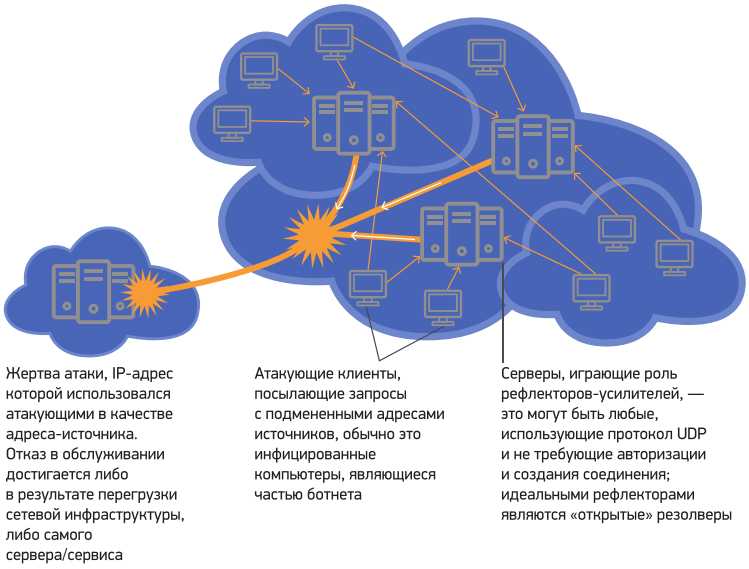

Рис. 23. Схема рефлекторной атаки с усилением.

Поскольку атакующий генерирует незначительный трафик с географически распределенного ботнета, этот трафик трудно идентифицировать и заметить что-либо подозрительное. В подавляющем большинстве случаев владельцы компьютеров ботнета сами не подозревают, что они являются источником атаки. А использование спуфинга адреса отправителя делает задачу определения источника практически невозможной.

С конца 1990-х годов DNS активно используется в качестве рефлектора и усилителя. Действительно, DNS является для этого идеальной услугой:

- DNS использует UDP, не требующий создания соединения и, таким образом, допускающий спуфинг;
- DNS является жизненно необходимой услугой, поэтому фильтрация входящего DNS-трафика практически не применяется;
- DNS является хорошим усилителем трафика, поскольку всегда можно найти запрос, размер ответа на который намного превышает размер самого запроса.

Это последнее свойство было известно с самого начала, однако ограничение максимального размера ответа 512 байт позволяло достичь лишь восьми-девятикратного усиления. Ситуация существенно изменилась с внедрением расширений EDNS0, позволяющих использовать дейтаграммы размером более

512 байт, и DNSSEC, включающего дополнительные данные. Если раньше для создания максимально возможного ответа использовались синтетические записи (например, специально созданная запись txt для определенного домена), то теперь стало возможно генерировать ответы большого размера для запросов, не вызывающих подозрения и использующих вполне безобидные домены.

Например, запрос всех существующих записей для имени isc.org с поддержкой DNSSEC

```
$ dig + dnssec isc.org. any
```

способен сгенерировать ответ размером около 3 килобайт, обеспечивая тем самым 50-кратное усиление (рис. 24).

Не будем забывать, что DNS — распределенная сеть с десятками, если не с сотнями миллионов серверов, обеспечивающих обработку запросов. Меньшая часть — это авторитетные серверы, отвечающие за определенные домены и связанные с ними записи, а подавляющее большинство составляют кеш-резолверы, обслуживающие весь рекурсивный процесс разрешения имен для клиента и предоставляющие последнему окончательный ответ.

Операционная практика предписывает авторитетным серверам отвечать только на запросы по доменам, которые они обслуживают, а резолверам — обрабатывать только запросы собственных клиентов — например, пользователей корпоративной сети. В реальности, к сожалению, зачастую авторитетные серверы также выполняют функцию резолверов, а резолверы готовы обслужить запрос любого клиента. Такие серверы получили название «открытых резолверов».

Открытые резолверы представляют серьезную проблему, поскольку являются идеальными рефлекторами-усилителями: они готовы обслужить запросы от любого клиента, их число колоссально, они повсеместны.

Решение этой проблемы, а точнее, «закрытие» резолвера обычно состоит в задании списков доступа, в который включаются сети локальных клиентов, для которых данный резолвер обеспечивает трансляцию имен. Рекомендации по обеспечению правильного функционирования резолвера опубликованы в RFC 5358[35]. Однако даже в этом случае при отсутствии мер по антиспуфингу, например, описанных в рекомендации BCP38 «Входная фильтрация: поражение атак отказа в обслуживании, использующих спуфинг IP-адресов источника»[36], остается возможность использования резолвера для атаки локальных клиентов.

---

[35] RFC 5358: Preventing Use of Recursive Nameservers in Reflector Attacks, URL: https://www.rfc-editor.org/rfc/rfc5358

[36] BCP38: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing, URL: https://www.rfc-editor.org/info/bcp38

```
$ dig +dnssec isc.org ANY
;; QUESTION SECTION:
;isc.org.                       IN      ANY

;; ANSWER SECTION:
dig +dnssec isc.org ANY
;; QUESTION SECTION:
;isc.org.                       IN      ANY

;; ANSWER SECTION:
isc.org.                7200    IN      RRSIG   SPF 5 2 7200 2013102
isc.org.                7200    IN      SPF     "v=spf1 a mx ip4:204.15
isc.org.                7200    IN      RRSIG   DNSKEY 5 2 7200 2013
isc.org.                7200    IN      RRSIG   DNSKEY 5 2 7200 2013
isc.org.                7200    IN      DNSKEY  257 3 5 BEAAAAOhHQD
isc.org.                7200    IN      DNSKEY  256 3 5 BQEAAAABwuH
isc.org.                3600    IN      RRSIG   NSEC 5 2 3600 20131026
isc.org.                3600    IN      NSEC    _adsp._domainkey.isc.org.
isc.org.                7200    IN      RRSIG   NAPTR 5 2 7200 20131026
isc.org.                7200    IN      NAPTR   20 0 "S" "SIP+D2U" "" _sip._
isc.org.                60      IN      RRSIG   AAAA 5 2 60 2013102620273
isc.org.                60      IN      AAAA    2001:4f8:0:2::69

...

;; ADDITIONAL SECTION:
asterisk.isc.org.       300     IN      A       149.20.32.15
mx.pao1.isc.org.        3600    IN      A       149.20.64.53
mx.pao1.isc.org.        3600    IN      AAAA    2001:4f8:0:2::2b
mx.ams1.isc.org.        3600    IN      A       199.6.1.65
mx.ams1.isc.org.        3600    IN      AAAA    2001:500:60::65
ams.sns-pb.isc.org.     7200    IN      A       199.6.1.30
ams.sns-pb.isc.org.     7200    IN      AAAA    2001:500:60::30
ord.sns-pb.isc.org.     7200    IN      A       199.6.0.30
ord.sns-pb.isc.org.     7200    IN      AAAA    2001:500:71::30
sfba.sns-pb.isc.org.    7200    IN      A       149.20.64.3
sfba.sns-pb.isc.org.    7200    IN      AAAA    2001:4f8:0:2::19
_sip._udp.isc.org.      7200    IN      RRSIG   SRV 5 4 7200 20131026202736 2013

...

;; Query time: 121 msec
;; SERVER: 199.6.1.30#53(199.6.1.30)
;; WHEN: Fri Sep 27 18:53:51 2013
;; MSG SIZE  rcvd: 3987
```

Рис. 24. Ответ на запрос всех существующих записей имени isc.org с поддержкой DNSSEC.

Response Rate Limiting (RRL) — ограничение частоты ответов

Если в случае резолверов проблему можно решить, оставив доступ только клиентам локальной сети, то для авторитетных серверов такой подход невозможен. По определению, они должны отвечать на запросы любого клиента. В этом случае негативный эффект можно уменьшить, применяя механизм ограничения частоты ответов, или RRL.

Этот механизм[37] был предложен Полом Викси (Paul Vixie) и Верноном Схряйвером (Vernon Schryver) и сначала внедрен в ПО BIND 9 (ISC), а впоследствии Knot DNS (CZ-NIC) и NSD (NLNetLabs).

Идея механизма проста: сервер отвечает только на ограниченное число запросов с идентичным ответом от одного и того же клиента. Ограничение определяется заданным администратором параметром — число ответов в секунду. Идентичными считаются ответы для одного и того же существующего доменного имени (QNAME) и типа записи (QTYPE). Также ответы на несуществующие поддомены (NXDOMAIN) или пустые запросы (NODATA) считаются идентичными и, соответственно, учитываются при подсчете.

Идентичными считаются клиенты, принадлежащие одной и той же сети (адресному блоку), размер которой задается администратором.

Этот механизм в первую очередь предназначен для авторитетных DNS-серверов. В случае рекурсивных резолверов применять его следует с большой осторожностью, чтобы не нарушить работу локальных клиентов. Дело в том, что ввиду недостаточного кеширования DNS-ответов многими приложениями повторяемость одинаковых запросов к резолверам от локальных клиентов достаточно велика, что может включить механизм ограничения. Например, при получении почтового сообщения сервер SMTP сделает запрос на записи NS, PTR, A и AAAA для входящего SMTP-соединения. Далее, при получении команды «Mail From» для этого же сообщения, сервер сделает дополнительные запросы для записей NS, A, AAAA, MX, TXT и SPF. Некоторые веб-браузеры также запрашивают одни и те же имена при обработке встроенных в веб-страницу изображений. Как мы уже говорили, наиболее правильным решением в этом случае является «закрытие» резолвера таким образом, чтобы он отвечал только на запросы, поступающие от локальных клиентов.

Механизм RRL имеет специальную возможность, позволяющую частично отвечать на запросы «нормальных» клиентов, адреса которых используются в попытке рефлекторной атаки. Вместо полного блокирования ответа на каждый второй запрос (этот параметр конфигурируется) сервер отвечает неполным, «обрезанным» ответом — небольшим пакетом с установленным флагом TC (truncation bit). После этого правильно функционирующий клиент должен сделать попытку получить полный ответ с помощью транспортного протокола TCP. Поскольку TCP требует установления соединения (включающего троекратный обмен данными между клиентом и сервером), для подложных адресов такая возможность исключена.

Использование технологии anycast

Когда DNS является объектом атаки, уменьшению рисков может помочь технология anycast (аникаст). В этом случае недостатки использования в DNS протокола UDP становятся преимуществами.

[37] https://kb.isc.org/docs/aa-01000

В октябре 2002 года система корневых серверов DNS подверглась по тем временам крайне массированной атаке DDoS. Объемы трафика достигали 100 Мбит/с, а суммарная мощность атаки превысила 900 Мбит/с. Трафик состоял из пакетов ICMP, TCP SYN, фрагментов TCP и UDP.

В результате атаки, которая продолжалась чуть больше часа, несколько корневых серверов оказались недоступны для большинства клиентов DNS. Интересно, что мощность самих серверов позволила бы им справиться с объемом запросов, но во многих случаях оказалась перегруженной связующая инфраструктура, что и привело к потерям трафика и отказу в обслуживании.

Атака 2002 года, хотя и имела незначительный эффект на производительность глобальной DNS, получила широкое освещение в прессе. Анализ атаки также послужил толчком для серьезного рассмотрения технологии аникаст в качестве способа распределения нагрузки. Аникаст позволяет уменьшить концентрацию трафика и локализовать атаку, создавая местные «точки притяжения».

Также технология аникаст позволяет решить задачу увеличения числа серверов, обслуживающих зону, без увеличения числа записей авторитетных серверов (записей NS) этой зоны, которое имеет свои пределы.

Во-первых, чем больше список серверов, тем больше размер ответа-реферала. Это, кстати, послужило причиной ограничения числа корневых серверов — 13, связанного с максимально допустимым размером сообщения в 512 байт, установленным стандартом DNS [RFC 1035 4.2.1]. Исторически это ограничение было вызвано максимальным размером пакета UDP, гарантирующим отсутствие фрагментации. И хотя сегодня расширение DNS EDNS0 (RFC 2671 2.3, 4.5) предусматривает предварительное соглашение о размере сообщения между клиентом и сервером, типичный размер пакета не превышает 4 кбайт.

Во-вторых, увеличение числа записей NS может негативно влиять на производительность системы. Например, в исследовании, проведенном Джефом Хьюстоном (Geoff Huston) с коллегами[38], было обнаружено, что некоторые резолверы при ошибке валидации DNSSEC повторяют эту попытку для всех существующих авторитетных серверов.

Существенно повысить устойчивость системы может применение технологии аникаст, известной с 1993 года, но ранее не использовавшейся в глобальном масштабе. Ее суть в том, что оператор анонсирует одну и ту же сеть (префикс IP и автономную систему) в различных частях Интернета. Благодаря архитектуре системы маршрутизации для любого клиента существует единственный и самый «короткий» путь к любой другой сети Интернета. Таким образом, аникаст позволяет клиенту установить связь с наиболее близкой в тополоти-

[38] https://www.potaroo.net/ispcol/2010-02/rollover.html

ческом смысле сетью без дополнительных изменений в ПО и протоколах! Принцип работы аникаст более подробно изложен в RFC 3258 «Распределение авторитетных серверов имен с использованием общего юникаст-адреса»[39], а схематично показан на рис. 25.

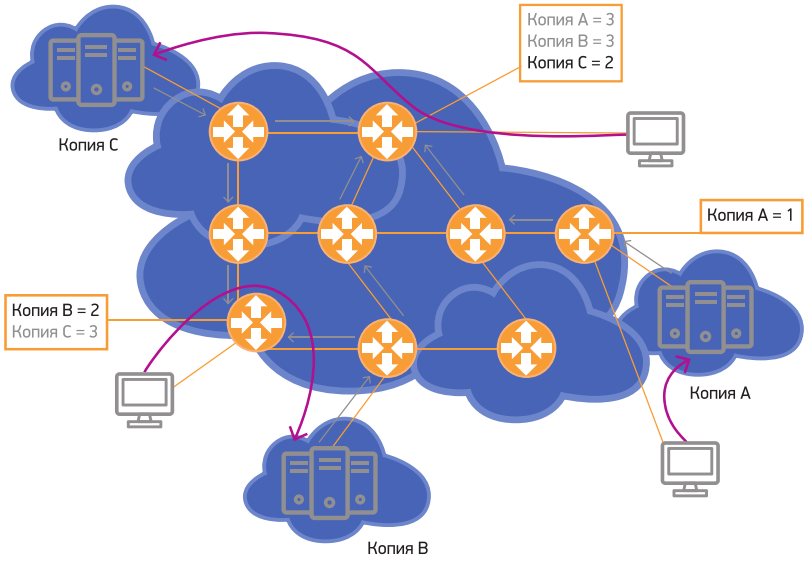

Рис. 25. Использование технологии аникаст для распределения нагрузки и повышения устойчивости DNS. Благодаря правилам выбора «лучшего пути» по протоколу динамической маршрутизации BGP каждый из клиентов получает доступ к «ближайшему» серверу DNS. Например, если «Копия A» и «Копия B» отстоят от клиента на три сетевых сегмента, а «Копия C» — на два, то в общем случае она и будет выбрана при маршрутизации трафика.

Технология аникаст наиболее подходит для приложений, которые используют протокол UDP, работающий без установления продолжительной связи. Например, при использовании TCP при каких-либо изменениях в топологии Интернета (которые происходят постоянно) кратчайший путь может измениться в процессе сеанса и привести клиента к другой сети аникаст, в результате чего связь будет разорвана.

В 2003 году после тщательной экспертной проверки и тестирования эта технология была впервые применена на корневом уровне DNS. Консорциум ISC, оператор корневого сервера f.root-servers.net (о корневых серверах мы более подробно скажем в разделе «Корневой уровень DNS»), разместил

[39] RFC 3258: Distributing Authoritative Name Servers via Shared Unicast Addresses, URL: https://www.rfc-editor.org/rfc/rfc3258

реплику своего сервера с использованием аникаст. Примеру ISC последовал ряд других операторов, и география системы КС существенно расширилась, как можно видеть на рис. 26. За период с 2003 по 2023 год число серверов выросло с изначальных 13 до более 1600.

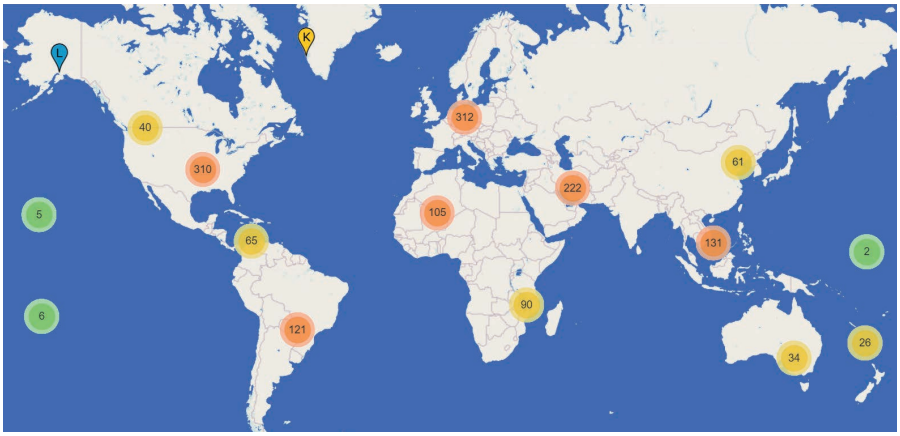

Рис. 26. Карта размещения корневых серверов DNS (апрель 2023 г.).
Источник: https://root-servers.org

Расширения безопасности DNS — DNSSEC

Теперь поговорим более подробно о технологии, позволяющей защитить многие уязвимые места DNS.

В рамках IETF были расширены возможности стандартного протокола DNS для решения проблемы аутентичности и целостности данных. Эти расширения получили название DNSSEC. Основная спецификация DNSSEC содержится в стандартах IETF RFC 4033[40], RFC 4034[41], RFC 4035[42] и RFC 5155[43]

DNSSEC позволяет пользователю убедиться, что полученные данные не были модифицированы в процессе публикации и передачи. Пользователь может быть уверен, что данные, содержащиеся в ответе на запрос DNS, в точности соответствуют данным в зоне для запрашиваемого доменного имени.

---

40 RFC 4033: DNS Security Introduction and Requirements,
URL: https://www.rfc-editor.org/rfc/rfc4033

41 RFC 4034: Resource Records for the DNS Security Extensions,
URL: https://www.rfc-editor.org/rfc/rfc4034

42 RFC 4035: Protocol Modifications for the DNS Security Extensions,
URL: https://www.rfc-editor.org/rfc/rfc4035

43 RFC 5155: DNS Security (DNSSEC) Hashed Authenticated Denial of Existence,
URL: https://www.rfc-editor.org/rfc/rfc5155

В рамках стандартного подхода к информационной безопасности рассматриваются три основных аспекта данных: их конфиденциальность, целостность и доступность. Целью технологии DNSSEC является защита целостности данных, а точнее — обеспечение возможности проверки целостности данных. Кстати, при использовании аникаст вопросы аутентичности и целостности данных встают еще острее, увеличивая ценность DNSSEC.

DNSSEC основан на криптографии с использованием открытых ключей. Администратор зоны подписывает все записи зоны своей цифровой подписью. Там же администратор публикует и открытый ключ, соответствующий этой цифровой подписи. Таким образом, путем проверки подлинности цифровой подписи и ее принадлежности администратору зоны пользователь может убедиться в целостности и аутентичности полученных данных.

Подлинность открытого ключа удостоверяет администратор родительской зоны путем включения в зону так называемой записи DS, представляющей собой хеш открытого ключа дочерней зоны. Разумеется, эта запись заверена цифровой подписью администратора этой родительской зоны. Его ключ, в свою очередь, удостоверяется администратором зоны верхнего уровня — и так далее, пока не будет достигнута так называемая точка доверия (trust anchor) — ключ, которому пользователь абсолютно доверяет. В DNSSEC таким ключом является ключ корневой зоны «.».

Таким образом, для проверки подлинности ответа на запрос DNS пользователю необходимо совершить целую цепочку проверок — от корневого ключа до ключа зоны, содержащей доменное имя. Этот процесс называется построением цепочки доверия. Только при успешном построении цепочки доверия и положительной проверке самой записи ответ может быть положительно удостоверен. Если хотя бы одна из проверок заканчивается неудачей, то и общий результат будет отрицательным.

К счастью, цепочка доверия DNSSEC полностью соответствует структуре делегирования имен, поэтому процесс построения цепочки доверия и разрешения имен происходит параллельно. На рис. 27 показан процесс разрешения имени с использованием DNSSEC.

Кстати, из рисунка видно, что в DNSSEC используются два типа ключей — так называемый ключ KSK (Key Signing Key, ключ подписи ключей) и ключ ZSK (ключ подписи зоны). Первый является ключом долговременного пользования и, соответственно, более сильным в криптографическом плане. Именно хеш ключа KSK «экспортируется» в родительскую зону в виде записи DS, устанавливая тем самым цепочку доверия. Ключ ZSK служит для подписания записей самой зоны. Ключ ZSK удостоверяется администратором домена путем подписи его ключом KSK. Поскольку в изменении ZSK задействован только администратор домена, этот процесс может (и должен) происходить достаточно часто.

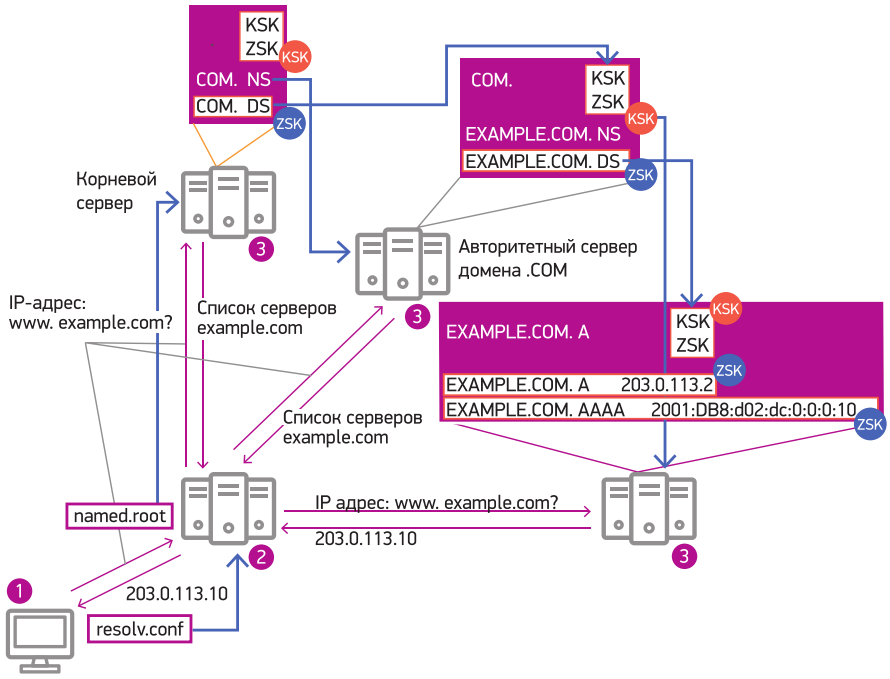

Рис. 27. Дополнительные элементы в процессе разрешения имен при внедрении DNSSEC.

Еще одна важная особенность DNSSEC — он позволяет удостовериться в несуществовании имени. Другими словами, при запросе разрешения несуществующего имени пользователю будет возвращен криптографически заверенный ответ, указывающий на отсутствие запрашиваемой записи в зоне. Для этого в DNSSEC используются записи NSEC и NSEC3[44]. Для того чтобы получить представление о значении различных записей DNSSEC, посмотрите на зону из примера в начале главы, но теперь сгенерированную с использованием DNSSEC (ниже). Как вы видите, размер зоны значительно увеличился за счет появления новых записей: RRSIG, DNSKEY, NSEC.

Запись RRSIG является основным компонентом DNSSEC, поскольку она содержит электронную подпись других записей ресурса, удостоверяющую их подлинность.

Публикация открытых ключей KSK и ZSK осуществляется с помощью записи DNSKEY. Подлинность ключей проверяется с помощью записи DS, которая находится в родительской зоне и соответствует ключу дочерней зоны. Для проверки пар

44 https://ru.wikipedia.org/wiki/DNSSEC

DS–DNSKEY в DNSSEC строится цепочка доверия, заканчивающаяся в корневой зоне (об этом мы подробнее поговорим в разделе «Подписание корневой зоны»).

Наконец, как мы уже говорили выше, записи NSEC и NSEC3 используются для удостоверения «несуществования имени». Для каждого существующего имени в зоне присутствует запись NSEC или NSEC3. Запись NSEC указывает, какие типы записей имеются для данного имени, а также на следующее имя в зоне. Недостатком NSEC является то, что с ее помощью можно легко определить все существующие имена, зарегистрированные в зоне. Чтобы не допустить этого, была разработана другая запись — NSEC3, в которой вместо явного имени используются хеши.

Таблица 4. Зона домена example.com, сгенерированная с использованием DNSSEC

Имя	TTL	Класс	Тип	Значение
$ORIGIN example.com.				
$TTL 86400				
@		IN	SOA	ns1.example.com. hostmaster.example. com. (
				2001062501 ; серийный номер зоны
				21600 ; обновлять каждые 6 ч
				3600 ; следующая попытка после 1 ч
				604800 ; срок годности 1 неделя
				86400) ; минимальный TTL 1 день
		IN	RRSIG	SOA 8 2 86400 20140627093132 20140530093 132 1443 example.com. (
				bmeJzAuSoYVAEynxoGtogoL710DZ/LrMLTib TEqm5rVWLsooTnMhIVb1ySS2paKB5xTskW bALou9jzwr6XRvRlEVc1YpYtMWhhEdrKYSG PRNCD/ShFi73LBVn7vwdqZL2sV6x+MyjfldT 64zNl45q56Z4yeSNEH96ni4qA5nVoY=)
		IN	NS	ns1.example.com. ; серверы имен, обслуживающие
		IN	NS	ns2.example.com. ; домен example.com
		IN	RRSIG	NS 8 2 86400 20140627093132 201405300931 32 1443 example.com. (
				kqbwys/cZBo5NbWyg+o5ygLruxk8Xzdg7Czg aVaU- nIfiXD3PpUGSDtoDgN5CJHioSoPHC7zz WrQ8ep6AZy4WJHySJT9M7Hkzo3CezotiTL EdWW24bBgrohmoPzhsshtiAEig/sTP9ywk+ +U3vfuwWZhWBaAOw6QF3f/WkCM9wdM=)

```
                IN      MX      10 mail1.example. com.          ; почтовые серве-
ры,
                                                                обслуживающие
I               N       MX      20 mail2.example. com.          ; домен
                                                                example.com
                IN      RRSIG   MX 8 2 86400 20140627093132 201405300931
                                32 1443 example.com. (

                                CTPgHC6fYu9wqMZn51r2FpmEO3wcx721rrG
                                9taLM4P5VZrZPHOAk2JiYFkLFGrDt21ecI6U
                                sr90nIDwc/00/qKZI6pzPQA3p31YER030Sopu
                                OIghVDOpgrI4AoleiyK5hQPlpqP7OoxDp3WH
                                MwI8U4JKr8Eqw8RkbAfvIiRtF9k=)

        3600    IN      DNSKEY 257 3 8 (                ; KSK

                                AwEAAb4WMOTBLTFvmBra5m6SK4VfViOz
                                mvyUAUoqv861ZQXeEFvwlndqNU9rwRsMx
                                rSWAYs5nHErKDn49usC/HyxxW1477iGFHh
                                fgL4mjNreJm9z  2QFB1VLbRbEPYdDMLCn
                                4cooqnG7/KG8W2i8Pym1L7f+aREwbLo+/71
                                6AS2PbaKMhfWLKLiq5wnBcUClQMNzCiwh
                                qxDJp1oePq
                                fkVdeUgXOtgiodYRIKyQFhJ5VWJ22npoi/ GifoXLCA-
                                DAlAwRLKc80/yJkCxskzgp

                                Hpw5Cki1lclgoaq4ssOuPRQ+ne6IHYCz9D2 mwzul-
                                blhLFamKdq7aHzNt4NlyxhpANVFi KLD8=
                                ); key id = 29332, size = 2048b 3600
                IN      DNSKEY 256 3 8 (                ; ZSK

                                AwEAAbd9WqjzE2Pynz21OG5doSf9hFzMr5d
                                hzz2waZ3vTa+005r7AjTAqmA1yH/B3+aAMih
                                Um5ucZSfVq07+kOaRE8yFj9aivOmA1n1+JLe
                                vJq/oyvQyjxQN2Qb89LyaNUT50KZIiL+uyyh
                                NW3KDR3SSbQ/GBwQNDHVcZi+JDR3RCor7
                                ); key id = 1443, size = 1024b
                IN      RRSIG   DNSKEY 8 2 3600 20140627093132 20140530
                                093132 29332 example.com. (

                                TozkmKFrlkoHt9y8vmioO12tWb1LhKsC64C3H
                                6+iDmsoQBYF9VDGYonDePw9lO4hZAWGaVN
                                fpEnrRQ+TRJrvyR5zNfxwca8D86jZ6ydKxpPll
                                2YshJxktsFcfdW6wdkaEhfwXSnkcIyYquIAQE
```

```
                                    B9TgmPoqTkFbgqLqW/4msbYQFcHqSetmoLju
                                    F+yqKoZnpWSt6eCodsGJnLzi3r1CcBytGSouz
                                    vjbhwZo3dhqEbWrY8MVg1H2Ybm1hAtcpOOqij
                                    a2hcF4rTyDxMau7RIzTGyFLLx1EoqtiOKHWbq
                                    2zzxj3VrEQZje4Uwimf9YSctTXk203km+Py2D
                                    WZWRWcoUaFQA==)

        86400   IN      NSEC    ftp.example.com. NS SOA MX RRSIG NSEC DNSKEY
                IN      RRSIG   NSEC 8 2 86400 20140627093132 20140530093
                                132 1443 example.com. (
                                fryQNKmBGDtIPomAwzgoPfRPFmTDoL+zAs9
                                d7BHrSk/y9uzmp9bPNS4Nob5np4vVbZOSH2
                                TgpEDon8vjUyPAnOZBZ8D9/riq6UQ6+EMUhD
                                zZ7PC1j323uLiLF2VuBznADuNGd2sjEZmLUi9 YToZ-
                                ZIAGEMpZNfOBUQW2IlhrC1ul=)
            ftp IN      A       203.0.113.12
                IN      A       203.0.113.13
                IN      RRSIG   A 8 3 86400 20140627093132 20140530093132
                                1443 example.com. (
                                mqL5M7SkyiLCDQFFQPcZB/P2zoq6Ezf9x51z/
                                LQuddtVldoliEA4C1quLh2rMwQwcPSuiReUct
                                SQ4LudKGsmB6noPTOUEmYYLT7nnJaMcqrm
                                2mGp5xUruGKPtkboaGjsabQFai/1/Stofz8EyR
                                ytfoJHMOGdUe+1xgPl7F4yVJQ=)
                IN      NSEC    mail1.example.com. A RRSIG NSEC
                IN      RRSIG   NSEC 8 3 86400 20140627093132 20140530093
                                132 1443 example.com. (
                                fd1h4zINeX+eAyRdTTIWyzZxn1DSXphhNUpb
                                Cllig6bljdhfMiKyVVz45f55q+GA8gB43+hBJ5j
                                2gj7bt3ytfYHi3Jp8ZWSTQUtTo+8NPfrBVunZ Yx-
                                sNWEerBAyVK8ZecVSjDq+87fyGbQjjX6oU kxai-
                                guVpFI97/6cKVKQ2ePo=)

mail1           IN      CNAME   server1
...
```

В то же время DNSSEC не является панацеей. Даже если полученные адреса серверов являются правильными, обмен данными может быть перехвачен или перенаправлен с использованием уязвимых мест системы маршрутизации (подробнее — в статье автора «Безопасность системы маршрутизации Интернета»). Также DNSSEC не обеспечивает шифрование данных, здесь необходима другая, довольно широко распространенная технология TLS (Transport Layer Security), которая использует цифровые сертификаты X.509. Именно на технологии TLS базируется протокол HTTPS. Наконец, DNSSEC не спасет от го-

мографии, когда имя сервера внешне очень похоже на другое имя, но на самом деле использует другие символы: например, цифра «1» и буква «l».

И все же использование DNSSEC совместно с другими средствами защиты существенно усиливает их эффективность. Например, в противоположность сертификатам доменных имен, используемых в TLS/HTTPS, цепь доверия в DNSSEC следует цепочке делегирования доменов и, таким образом, основана на деловых отношениях, существующих при регистрации доменов. Об этом мы более подробно поговорим в разделе «Усиление безопасности других услуг с помощью DNS».

Вопросы внедрения DNSSEC

Внедрение DNSSEC является комплексной задачей, требующей участия многих сторон — администраторов и операторов доменных зон и в первую очередь национальных доменов, а также регистраторов, провайдеров хостинга, сетевых провайдеров и разработчиков программного обеспечения. Подобно другим новым технологиям, преимущество использования DNSSEC зависит от степени ее внедрения. Увы, порой образуется замкнутый круг — держатели доменных имен не видят смысла инвестиций во внедрение DNSSEC, пока значительная часть клиентов не станет осуществлять валидацию. А клиенты, в свою очередь, ожидают более масштабного распространения DNSSEC в дереве DNS.

Дополнительные затраты

Да, применение DNSSEC имеет значительные преимущества, но с внедрением этой технологии связаны дополнительные затраты и трудности.

Увеличение размера файла-зоны. Как мы уже видели, использование DNSSEC требует создания дополнительных записей в зоне. Основной вклад в размер зоны вносят записи NSEC (NSEC3) для проверки несуществования имени и запись RRSIG, являющаяся цифровой подписью других записей. Увеличение размера зоны варьируется в зависимости от ее содержимого, размеров используемых ключей, метода подписания, но может превышать восемь крат.

Увеличение размера ответов. Поскольку каждый ответ содержит цифровую подпись, размер ответов также увеличивается в несколько раз. Большой размер ответов может также усилить ущерб от возможной DDoS-атаки, когда множество небольших запросов DNS, посланных с использованием подложного IP-адреса жертвы (и якобы от его имени), вызывают ответный трафик, значительно превышающий входящий. Об этом мы поговорим в следующем разделе.

Увеличение числа запросов. По мере внедрения DNSSEC можно ожидать некоторое увеличение числа дополнительных запросов на получение ключей и построение цепочек доверия.

Большая нагрузка на «клиента». Помимо разрешения имен, клиенту необходимо производить проверку подписей, осуществлять построение цепочек доверия.

Потребуется дополнительная вычислительная мощность, хотя при сегодняшнем развитии технологии это вряд ли является существенной проблемой.

Усложнение инфраструктуры и процесса генерирования зоны. Внедрение DNSSEC требует дополнительной инфраструктуры для генерирования, хранения и обновления ключей, подписания зоны, а также для связанных с этими операциями технических и административных процессов. В дополнение DNSSEC вводит в DNS понятие абсолютного времени, за точностью которого необходимо следить.

Применение DNSSEC на пользовательском уровне

DNSSEC является достаточно зрелой технологией, внедренной в значительном числе доменов верхнего уровня (на июль 2023 года - более 90% всех доменов верхнего уровня[45]). Тем не менее, использование DNSSEC на пользовательском уровне пока незначительно. А ведь именно это определяет эффективность защиты, которую обеспечивает технология DNSSEC.

Существует несколько задач, решение которых необходимо для более широкого распространения DNSSEC на пользовательском уровне.

Внедрение DNSSEC для доменных имен второго и третьего уровня

Если для доменов верхнего уровня ситуация более или менее благополучна, то среди доменных имен второго и последующих уровней DNSSEC практически не внедрен. Очевидно, что для получения эффекта от этой технологии DNSSEC должен быть поддержан по всей цепочке доверия — от собственно запрашиваемого доменного имени до корня DNS.

Подписание корневой зоны явилось хорошим стимулом для операторов национальных и других доменов верхнего уровня. В свою очередь, подписание национального домена и поддержка безопасной делегации — то есть записей DS, удостоверяющих открытые ключи дочерних зон, — стимулирует держателей доменов нижних уровней к внедрению DNSSEC.

Однако этого все же недостаточно. Необходима также административная поддержка DNSSEC на уровне регистраторов. Как минимум нужно обеспечить поддержку создания записи DS в родительской зоне. Но для практического применения эта процедура должна быть существенно упрощена, как, например, это делают GoDaddy[46] и Cloudflare[47].

Для широкомасштабного внедрения DNSSEC также необходима поддержка этой технологии провайдерами хостинга, особенно если последние не являются и регистраторами доменных имен. Поскольку клиентами провайдеров, как

45 https://rick.eng.br/dnssecstat
46 https://uk.godaddy.com/help/turn-dnssec-on-or-off-6420
47 https://developers.cloudflare.com/dns/dnssec

правило, являются простые пользователи, процедура включения DNSSEC должна быть максимально упрощена и автоматизирована, включая создание и обслуживание ключей, подписание и безопасное делегирование. Например, шведский хостинг-провайдер Binero[48] автоматически подписывает все домены, хостинг которых он предоставляет. Таким образом, для включения DNSSEC пользователю не приходится делать практически ничего.

Поддержка DNSSEC резолверами сервис-провайдеров и корпоративных сетей

Как бы широко ни был внедрен DNSSEC в системе DNS, без проверки подлинности ответов на пользовательском уровне эффект от этой технологии сводится к нулю. Данная задача может быть решена поэтапно.

Наиболее простой и в то же время эффективный шаг — обеспечение поддержки DNSSEC резолверами Интернета сервис-провайдеров и корпоративных сетей. Поскольку разрешение имен для пользователей сетей доступа и корпоративных сетей осуществляют именно эти резолверы, проверка подлинности ответов позволит усилить защиту DNS для широкой клиентской базы. Безусловно, слабым звеном в этой схеме остается сегмент сети между резолвером и пользовательским приложением (например, веб-браузером и почтовым приложением), но, как правило, этот сегмент находится в зоне административного контроля провайдера и может быть эффективно защищен от потенциальных угроз, которые рассматривались выше. Примером такого подхода является американский провайдер широкополосного доступа Comcast[49], внедривший в январе 2012 года DNSSEC в своей инфраструктуре резолверов, что обеспечило защиту DNSSEC для 17,8 миллиона пользователей. Большинство сетевых сервис-провайдеров Швеции также осуществляют проверку имен для своих пользователей.

По состоянию на июль 2023 года, согласно статистике APNIC Labs[50], более 30% пользователей в мире производят валидацию ответов с помощью DNSSEC. Значительный вклад здесь вносит «публичный» резолвер Google — Public DNS, который обслуживает 14% запросов с DNSSEC. Как видно из рис. 28, процент проникновения DNSSEC на пользовательском уровне существенно различается для стран и сетей.

Поддержка DNSSEC на уровне приложений

Полноценной моделью «сквозного» внедрения DNSSEC, которая может рассматриваться как следующий шаг, является проверка подлинности ответа самим приложением или операционной системой. При этом приложение может по-прежнему производить разрешение имени через резолвер сервис-провайдера, но последний вместо проверки подлинности полученных

[48] https://binero.com
[49] http://www.comcast.com
[50] https://stats.labs.apnic.net/DNSSEC

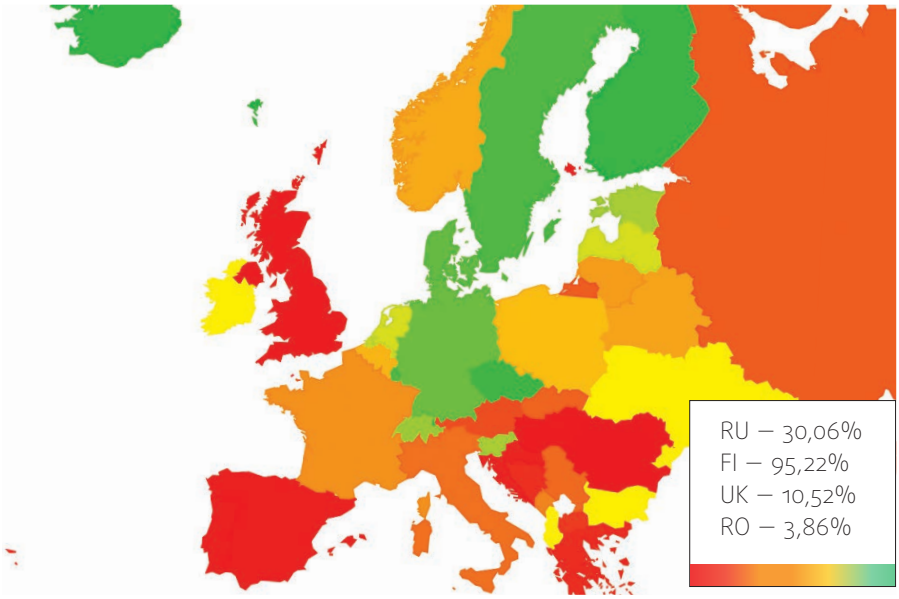

Рис. 28. Карта использования DNSSEC европейскими пользователями Сети. (данные на февраль 2024 г.)
Источник: https://stats.labs.apnic.net/dnssec/XE

ответов и выдачи окончательного результата будет передавать «сырые» данные (то есть включающие записи RRSIG, NSEC/NSEC3 и DNSKEY) непосредственно приложению. Таким образом задача итеративного разрешения имени и построения цепочки доверия ложится на пользовательское приложение (или операционную систему), а резолвер используется в целях кеширования и оптимизации трафика. Примером реализации такого подхода является программа DNSSEC-Trigger[51], работающая совместно с установленным на компьютере пользователя локальным итеративным резолвером Unbound[52] от компании NLnet Labs.

Защита конфиденциальности в DNS

При разработке DNS основным требованием с точки зрения безопасности была доступность. Позже стало очевидно, что целостность данных DNS очень важна, и для защиты данных DNS при передаче были разработаны стандарты DNSSEC. Но до недавнего времени достижение конфиденциальности никогда не было целью. В конце концов, данные DNS общедоступны по определению.

Однако, хотя сами данные DNS являются общедоступными, связанные с ними метаданные и, в частности, данные о том, кто и что именно запрашивает, рассматриваются как серьезная угроза конфиденциальности пользователей.

[51] https://nlnetlabs.nl/projects/dnssec-trigger/about
[52] https://nlnetlabs.nl/projects/unbound/about

Существует два основных подхода к решению проблем конфиденциальности DNS. Во-первых, затруднить прослушивание DNS-запросов с помощью шифрования DNS-транзакций. Это поддерживают два основных механизма — DNS-over-TLS (DoT) и DNS-over-HTTP (DoH). Второй подход заключается в уменьшении раскрытия информации за счет уменьшения объема информации в каждом DNS-запросе.

Эти механизмы предоставляют пользователю более высокий уровень защищенности частной информации, поскольку клиентские DNS-запросы становятся невидимыми для пассивного наблюдателя. Однако стоит помнить, что вероятность утечки все же есть, так как ваши запросы и ответы на них известны не только вам. В системе, где между вами и авторитетными DNS-серверами есть посредники (резолверы), абсолютной конфиденциальности быть не может. Даже если вы отправляете запросы на авторитетные DNS-серверы напрямую, они все равно будут знать, что вы запросили у них эту информацию. Поэтому имеет смысл внимательно ознакомиться с политикой конфиденциальности вашего поставщика услуг DNS, прежде чем сделать выбор.

DNS поверх TLS

DNS поверх TLS (DoT) задокументирован в RFC 7858[53]. Он использует протокол Transport Layer Security (TLS) для шифрования связи между клиентом и сервером, а также для того, чтобы клиент мог аутентифицировать сервер. Во многом так же, как TLS используется для защиты сеансов HTTP и обеспечения некоторой уверенности в том, что сервер является авторизованным агентом указанной службы, этот протокол также может использоваться в контексте DNS между DNS-клиентом (обычно это резолвер-заглушка операционной системы) и выбранным рекурсивным резолвером, поддерживающим DoT.

DNS поверх HTTPS

DNS поверх HTTPS (DoH) — это стандарт, задокументированный в RFC 8484[54]. Он похож на DoT тем, что позволяет шифровать все DNS-запросы, так что только DNS-клиент (обычно это браузер пользователя) и сервер DoH по вашему выбору (обычно рекурсивный резолвер, поддерживающий DoH) знают, на какие сайты собирается перейти пользователь. Описанный подход — это больше, чем туннель через HTTP. Он устанавливает типы форматирования мультимедиа по умолчанию для запросов и ответов, но использует обычные механизмы согласования содержимого HTTP для выбора альтернатив, которые клиенты могут предпочесть в будущих вариантах использования. В дополнение к этому согласованию типа мультимедиа он согласуется с функциями HTTP, такими как кеширование, перенаправление, проксирование, аутентификация и сжатие.

---

53 RFC 7858: Specification for DNS over Transport Layer Security (TLS), URL: https://www.rfc-editor.org/rfc/rfc7858

54 RFC 8484: DNS Queries over HTTPS (DoH), URL: https://www.rfc-editor.org/rfc/rfc8484

Хотя DoH достигает цели защиты данных пользовательских запросов от прослушивания третьими сторонами, его внедрение вызвало озабоченность, в основном связанную с моделью развертывания, а не с самим протоколом.

Во-первых, DoH очень сложно обнаружить. Он выглядит как HTTPS-трафик и использует тот же порт, что и HTTPS-трафик. Возможность обнаружения DoH путем проверки имени сервера, которое передается в открытом виде во время «рукопожатия» TLS, исчезает с работой над зашифрованным SNI в TLS 1.3. Это ломает многие существующие реализации, от родительского контроля до инструментов, развернутых интернет-провайдерами для соблюдения различных правил. Но опять же — эта проблема возникает только в определенных сценариях развертывания. Например, если резолвер DNS, выбранный пользовательскими приложениями, является резолвером DNS интернет-провайдера, который поддерживает DoH, данные DNS видны интернет-провайдеру и DoH не представляет никакой проблемы.

Это приводит ко второй проблеме — выбору сервера DoH. Вместо использования локально настроенной службы резолвера DNS, предоставляемой интернет-провайдером (например, через запрос DHCP), текущие реализации предписывают использовать службу, настроенную браузером. Например, конфигурация DoH в Firefox предоставляет резолвер DNS Cloudflare по умолчанию, но позволяет пользователю самостоятельно выбрать доверенный рекурсивный резолвер.

DNS поверх QUIC

Хотя DoT и DoH обеспечивают защищённость данных между DNS-клиентом и рекурсивным резолвером, оба протокола используют TCP в качестве транспорта. Помимо значительных накладных расходов (по сравнению с UDP, основным транспортным протоколом DNS), TCP может являться причиной низкой производительности, особенно когда используется многопоточная передача (как в случае HTTPS/2). Дело в том, что при потере сегмента данных TCP будет ожидать повторной передачи этого сегмента, тем самым блокируя получение остальных данных, возможно, относящихся к другому логическому потоку.

Решением этой проблемы, в частности, для HTTPS стал стандартизованный в 2021 году протокол QUIC[55]. Хотя этот протокол был разработан в основном для поддержки HTTPS, QUIC является транспортным протоколом общего применения, и DNS поверх QUIC, или DoQ[56], тому свидетельство.

Основным преимуществом использования QUIC по сравнению с TCP является то, что он использует UDP в качестве базового транспорта, обеспечивая целостность передаваемых данных и мультиплексирование на уровне приложений. Это значительно повышает производительность QUIC, и как следствие – DoQ.

---

55 RFC 9000: A UDP-Based Multiplexed and Secure Transport, URL: https://www.rfc-editor.org/rfc/rfc9000

56 RFC 9250: DNS over Dedicated QUIC Connections, URL: https://www.rfc-editor.org/rfc/rfc9250

Минимизация имен Qname

В настоящее время, когда преобразователь получает запрос «Какова запись AAAA для www.example.com?», он отправляет тот же вопрос на все авторитетные серверы, начиная с корневых серверов (см. рис 18. Процесс трансляции имен в DNS). Но отправка полного имени запроса (www.example.com) авторитетному серверу имен является традицией, а не требованием протокола.

В рамках рабочей группы DNSOP IETF был разработан подход, стандартизованный в RFC 9156 «Минимизация имен запросов DNS для улучшения конфиденциальности»[57]. Идея состоит в том, что каждый авторитетный сервер получает только ту часть общего вопроса, на которую он уполномочен отвечать. Например, от корневых серверов преобразователь ожидает узнать имя авторитетных серверов для домена .com, поэтому вместо отправки вопроса «www.example.com.» достаточно запросить информацию (имена авторитетных серверов) о домене «com.».

Усиление безопасности других услуг с помощью DNS

DANE (DNS-based Authentication of Named Entities — Аутентификация поименованных объектов с использованием DNS)

Как мы уже заметили, внедрение DNSSEC также открывает новые возможности. Например, решение проблем, связанных с сертификатами открытых ключей (X.509), используемых для проверки достоверности веб- или почтовых серверов и обмена информацией с использованием защищенного протокола TLS. Эта система обобщенно называется Web-PKI.

Дело в том, что выдачей этих X.509 сертификатов занимаются несколько сотен независимых удостоверяющих центров (УЦ), которые используют данные третьих сторон (например, данные от компаний-регистраторов) для проверки прав пользования доменным именем. Конкретный список доверенных УЦ определяется производителями веб-браузеров – такими компаниями, как Google, Apple, Mozilla Foundation и т.д. Хотя в выборе УЦ они руководствуются требованиями CA/Browser Forum[58], а также результатами аудита WebTrust Task Force[59] и ETSI[60], в основном этот процесс закрыт и непрозрачен.

В Web-PKI все УЦ, корневые сертификаты которых установлены в браузере пользователя, имеют право выдавать сертификаты для любого доменного имени и при этом имеют одинаковый уровень доверия. Любой из этих корневых сертификатов является так называемой точкой доверия (Trust Anchor, TA). Это значит, что УЦ с

[57] RFC 9156: DNS Query Name Minimisation to Improve Privacy, URL: https://www.rfc-editor.org/rfc/rfc9156
[58] https://cabforum.org/about-the-baseline-requirements
[59] https://www.cpacanada.ca/en/business-and-accounting-resources/audit-and-assurance/overview-of-webtrust-services/principles-and-criteria
[60] https://portal.etsi.org/TB-SiteMap/ESI/Trust-Service-Providers

недостаточным качеством проверок прав владения именем является «слабым звеном» всей системы. Злоумышленник может использовать такой УЦ для получения конкурирующего сертификата для уже существующего чужого имени или сертификата для несуществующего имени. Даже флагманы индустрии оказываются уязвимы – так, компания Symantec на протяжении нескольких лет генерировала сертификаты для несуществующих имен, с просроченным сроком действия и другими нарушениями требований CA/Browser Forum[61].

Дело усугубляется еще и тем, что корневые УЦ часто делегируют полномочия выдачи сертификатов подчиненным УЦ. В результате в Интернете существует большое количество доверенных УЦ, качество регистрационных процессов которых проверить практически невозможно.

Исправить ситуацию можно, если владелец доменного имени будет сам контролировать выдачу сертификатов, а по возможности вообще не будет пользоваться услугами третьих лиц.

Здесь на помощь приходит DNS. Действительно, если владелец имени или доменной зоны контролирует публикацию такой важной информации, как IP-адреса сайтов, почтовых серверов, обслуживающих домен, и т.п., почему бы не создать новую запись, с помощью которой владелец смог бы указать на сертификат, который следует использовать при обращении к сайту или почтовому серверу, связанным с этим доменным именем? Разумеется, запись должна быть защищена, и DNSSEC выполняет эту функцию.

Разработка и стандартизация соответствующих протоколов была выполнена в рамках рабочей группы IETF DANE[62]. DANE позволяет владельцу домена указать, какой сертификат TLS/SSL должно использовать приложение или служба для подключения к вашему сайту. Для этого используется запись TLSA, стандартизованная в RFC 6698[63] и RFC 7671[64].

Основой DANE является новая запись ресурса TLSA. Эта запись с доменным именем хоста содержит инструкции по использованию TLS/SSL-сертификата при доступе к сервису по указанному порту и протоколу. Так, например, запись ниже указывает, что при валидации сертификата, полученного при обращении к веб-серверу www.example.com по протоколу HTTPS (порт 443, протокол TCP), необходимо использовать точку доверия (trust anchor, TA), указанную в последнем поле записи:

---

61 https://wiki.mozilla.org/CA/Symantec_Issues

62 https://datatracker.ietf.org/wg/dane/documents/

63 RFC 6698: The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA, URL: https://www.rfc-editor.org/rfc/rfc6698

64 RFC 7671: The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance, URL: https://www.rfc-editor.org/rfc/rfc7671

_443._tcp.www.example.com. IN TLSA 2 0 1 (E8B54E0B4BAA815B06D3462D65FBC7C0CF5 56ECCF9F5303EBFBB77D022F834C0)

Поля после метки TLSA имеют следующее значение:

IN TLSA	2	0	1	E8B54E0B4BAA8...
	Использование сертификата	Селектор	Тип сравнения	Данные, ассоциированные с сертификатом
	0 – PKIX-TA 1 – PKIX-EE 2 – DANE-TA 3 – DANE-EE	0 – использовать весь сертификат для сравнения 1 – использовать только открытый ключ для сравнения	0 – все данные, указанные селектором, используются для сравнения 1 – хеш SHA-256 данных используются для сравнения 2 хеш SHA-512 данных используются для сравнения	Данные, необходимые для сравнения, закодированные в виде строки BASE64

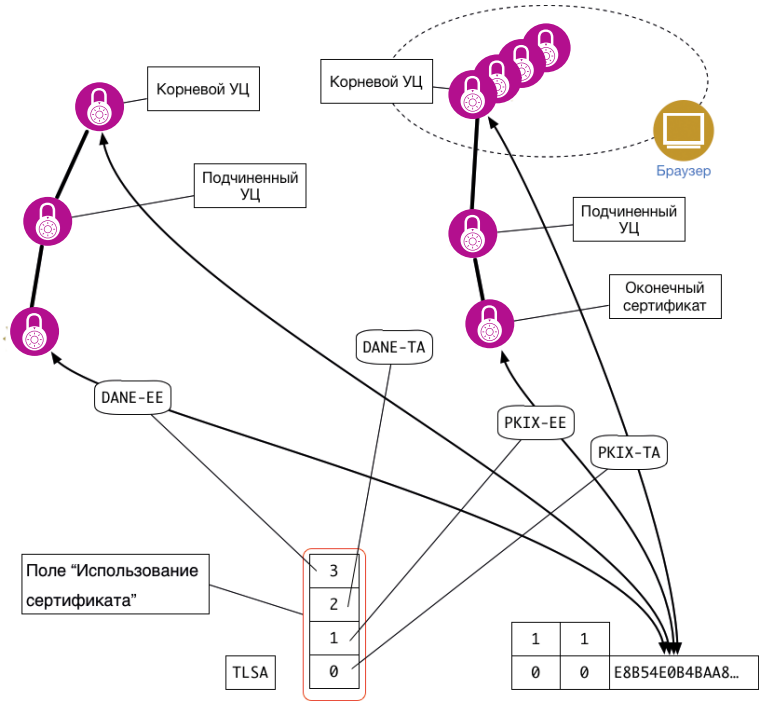

Рис. 29. Различные типы записи TLSA и связанные с ними проверки.

Значение 0 (PKIX-TA) указывает, что путь валидации полученного сертификата (или открытого ключа в зависимости от значения поля «Селектор») должен проходить через сертификат, указанный в поле «Данные, ассоциированные с сертификатом». Это может быть как корневой, так и подчиненный сертификат, но в любом случае путь валидации должен заканчиваться точкой доверия, зарегистрированной в пользовательском браузере.

Значение 1 (PKIX-EE) накладывает ограничение на сертификат, полученный от веб-сервера. Он (или его открытый ключ) должен соответствовать «данным, ассоциированным с сертификатом». При этом сертификат должен пройти проверку, используя путь валидации к одной из точек доверия браузера.

Значение 2 (DANE-TA) применяется для определения сертификата, который должен использоваться в качестве новой точки доверия при валидации сертификата, полученного от веб-сервера. Этот метод также называют «утверждением точки доверия», поскольку он позволяет владельцу доменного имени указать точку доверия, которая не входит в стандартную коллекцию TA пользовательского браузера.

Наконец, значение 3 (DANE-EE) определяет метод, когда сертификат (или его открытый ключ), полученный от сервера, должен совпадать с сертификатом, указанным в записи TLSA. Этот метод позволяет владельцу доменного имени использовать собственные сертификаты без привлечения сторонних УЦ. Этот метод отличается от PKIX-EE тем, что он не требует дополнительной валидации сертификата через путь доверия.

DANE может использоваться не только при взаимодействии с веб-серверами. Например, защита передачи данных между почтовыми серверами является чрезвычайно важной. При этом часто используется протокол TLS, но аутентичность сертификата сервера представляет еще большую проблему, чем в случае с Web-PKI. DANE и здесь окажет администраторам почтовых серверов неоценимую услугу, делая почтовую систему в целом более защищенной. Более подробно о проблематике защиты передачи почтовых сообщений между транспортными почтовыми агентами (Mail Transport Agent, MTA) и о применении DANE в этом случае для защиты протокола SMTP описано в RFC 7672[65].

Другие протоколы приложений, например, IMAP или XMPP, могут также использовать DANE. Обсуждение использования DANE в этих случаях можно найти в RFC 7673[66].

---

[65] RFC 7672: SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS), URL: https://www.rfc-editor.org/rfc/rfc7672

[66] RFC 7673: Using DNS-Based Authentication of Named Entities (DANE) TLSA Records with SRV Records, URL: https://www.rfc-editor.org/rfc/rfc7673

Другими словами, DNSSEC+DANE обладают существенным потенциалом обеспечения защищенности коммуникационной инфраструктуры Интернета на уровне приложений.

DKIM, SPF и DMARC

Одна из проблем существующей инфраструктуры электронной почты заключается в том, что отправитель может использовать любое доменное имя для различных идентификаторов в заголовке почты, а не только собственное доменное имя. Эта уязвимость используется спамерами, чтобы ввести пользователей в заблуждение, подменив домен отправителя. Sender Policy Framework (SPF, Основа политики отправителя)[67] — стандартизированный метод предотвращения подделки адреса отправителя. SPF позволяет администраторам указать, каким хостам разрешено отправлять почту от имени данного домена, путем создания специальной записи SPF в DNS. Почтовые обменники используют эту запись DNS для проверки того, что почта с данного домена отправляется хостом, санкционированным администратором этого домена.

Хотя SPF гарантирует, что почта может поступать только с авторизованных почтовых серверов, он не защищает само сообщение электронной почты, включая его заголовки, такие как «От:», «Кому:», «Дата:». Это означает, что спамер все еще может выдать себя за кого-то, подделав заголовок «От:», и это то, на что пользователь обратит внимание. DomainKeys Identified Mail (DKIM, Идентификация почты с помощью DomainKeys)[68] использует асимметричную криптографию для цифровой подписи сообщения. DKIM возьмет хеш нескольких полей электронной почты, в том числе «От:», «Кому:», «Дата:». Затем этот хеш подписывается закрытым ключом, который генерируется администратором домена и помещается в заголовок DKIM. Открытый ключ домена публикуется в DNS для этого домена и используется для проверки подлинности электронной почты

DKIM не защищает от подделки поля «От:» напрямую, но гарантирует, что электронное письмо действительно пришло из рассматриваемого домена. Например, DKIM может гарантировать, что электронное письмо пришло из домена example.com, но не обязательно может гарантировать, от кого в этом домене отправлено сообщение, поскольку ключ используется для всего домена, а не для отдельных отправителей.

Domain-based Message Authentication, Reporting and Conformance (DMARC, Аутентификация, отчетность и определение соответствия сообщений на основе доменного имени)[69] позволяет владельцу домена публиковать политики обработки сообщений

---

[67] RFC 7208: Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1, URL: https://www.rfc-editor.org/rfc/rfc7208

[68] RFC 6376: DomainKeys Identified Mail (DKIM) Signatures, URL: https://www.rfc-editor.org/rfc/rfc6376

[69] RFC 7489: Domain-based Message Authentication, Reporting, and Conformance (DMARC), URL: https://www.rfc-editor.org/rfc/rfc7489

для получателей сообщений электронной почты, исходящих из этого домена, и запрашивать отчеты об аутентификации полученной почты. Политика DMARC позволяет отправителю указать, что его сообщения защищены с помощью SPF и/или DKIM, и сообщает получателю, что делать, если ни один из этих методов проверки подлинности не проходит – например, поместить сообщение в карантин или отклонить. DMARC также позволяет получателю электронной почты сообщать отправителю о сообщениях, прошедших и/или не прошедших оценку DMARC.

Эти политики публикуются в DNS в виде записей TXT.

Координация и администрирование доменных имен верхнего уровня

Общий взгляд на систему. Структуры ICANN

Координацию глобальной системы имен, а точнее, корневого уровня DNS, осуществляет ICANN (Internet Corporation for Assigned Names and Numbers) — частная некоммерческая корпорация, зарегистрированная в штате Калифорния, США. Задачу координации корневого уровня DNS можно разделить на две части: «что» может быть включено в корневую зону в качестве имени, а также «как» — процедурные и операционные вопросы включения и обслуживания этого имени. Первый вопрос, «что», принадлежит уровню разработки политик и правил. Второй же, «как», — вопрос исполнения принятых политик и правил, внесения изменений в корневую зону и ее обслуживания.

Организационные структуры, созданные для решения этих вопросов, различны.

В структуре ICANN существуют две так называемые организации поддержки — организация поддержки общих имен gNSO (Generic Names Supporting Organization) и организация поддержки национальных доменных имен ccNSO (Country Code Names Supporting Organisation). Они занимаются процедурными аспектами и обеспечивают разработку политик.

Разработка политик gNSO происходит согласно утвержденному процессу разработки политик (Policy Development Process, PDP). Процесс предусматривает работу над определением проблемы, а при разработке решения опирается на широкую поддержку общественности и обратную связь, также используя ресурсы консультативных комитетов: ALAC, GAC, RSAC и SSAC. Принятие политик происходит на основе консенсуса, а ратификация производится Советом ICANN.

Работу gNSO координирует совет, в состав которого входят 22 члена, номинированные так называемыми заинтересованными группами (также именуемые стейкхолдерами, от английского stakeholder) — реестрами, регистраторами, коммерческими и некоммерческими группами. Такая широкая избирательная база позволяет сбалансированно отразить интересы различных сторон.

Чтобы получить представление о типе вопросов, рассматриваемых gNSO, достаточно посмотреть на некоторые принятые политики— «Единая политика разрешения споров о доменных именах», «Политика удаления доменов с истекшим сроком действия», «Политика изменения регистраторов».[70]

Как следует из названия, ccNSO занимается вопросами национальных доменов. Некоторые из них весьма щепетильны, поскольку граничат с национальными интересами государств. Хотя PDP ccNSO по структуре похож на PDP gNSO, участие комитета GAC (Government Advisory Committee, Правительственный консультативный комитет) в нем прописано более явно.

Процесс делегирования и ре-делегирования национальных доменов верхнего уровня описан в документе IANA «Delegating or redelegating a country-code top-level domain (ccTLD)»[71]. Основным методом определения допустимости того или иного доменного имени является включение соответствующего «alpha-2» кода таблицы ISO 3166-1. Другим методом является утверждение имени через так называемый ускоренный процесс рассмотрения национальных IDN-доменов верхнего уровня (IDN ccTLD Fast Track Process).

Многие правила и политики, разработанные и утвержденные вышеперечисленными группами и комитетами ICANN, влияют на содержимое корневой зоны. Они определяют процессы внесения изменений, а также осуществление обслуживания корневой зоны DNS — об этом мы поговорим в следующих разделах.

Корневой уровень DNS

Корневая зона содержит информацию обо всех доменах верхнего уровня: национальных доменах (например, .ru, .рф), доменах общего назначения (например, .com, .museum, .москва, .дети[72]). Точнее, эта информация содержит списки серверов имен, обслуживающих тот или иной домен верхнего уровня. Таким образом клиенту указывается, на какие серверы DNS отправить последующий запрос для продолжения разрешения полного доменного имени. Как мы видели, говоря о процессе разрешения имени, любой «свежий» (то есть не сохраненный в кеше резолвера) запрос начинается с обращения к так называемым корневым DNS-серверам (КС), обеспечивающим доступ к зоне.

Мы не напрасно отметили, что запрос является «свежим». Дело в том, что обычно резолвер запоминает ответы, полученные от серверов DNS, и на по-

[70] https://www.icann.org/consensus-policies-ru

[71] https://www.iana.org/help/cctld-delegation

[72] Сегодня домены общего назначения (generic top-level domains, gTLD) включают исторически общие домены, такие как .com и .net, спонсированные домены (.aero, .coop), географические домены (.cat, .asia), а также домены, созданные в рамках начатой в 2008 г. программы ICANN по масштабному созданию доменов верхнего уровня.

вторные запросы отвечает данными из своего кеша. Время хранения ответов определяется администратором соответствующего домена и в случае корневой зоны для большинства записей равняется 48 часам.

Система корневых серверов и координация ее работы

Генерация и распределение корневой зоны

Состав корневой зоны постоянно меняется. В среднем в зону вносится несколько изменений в неделю. Например, изменяется информация о DNS-серверах, обслуживающих домен верхнего уровня. Или же требуется добавление нового домена верхнего уровня, хотя такие изменения происходят гораздо реже. Как это происходит?

Запрос на изменение поступает от администратора домена верхнего уровня (ccTLD, gTLD и т.д.) и обслуживается IANA (Internet Assigned Numbers Authority) — структурой, отвечающей за регистрацию изменений в корневой зоне, оператором которой является ICANN[73].

После проведения необходимых административных и технических процедур (например, проверки правильности и законности запроса, проверки возможных негативных последствий для корневой зоны) запрос на изменение подписывается цифровым образом и направляется в организацию, ответственную за публикацию зоны в DNS, называемой Root Zone Maintainer (организация по обслуживанию корневой зоны). Эту роль в настоящее время выполняет компания VeriSign по контракту с ICANN[74].

Зона публикуется на скрытом мастер-сервере и затем распространяется на все корневые серверы с использованием протокола TSIG, защищающего данные от модификации при передаче. Независимо от наличия или отсутствия изменений корневая зона обновляется дважды в день.

Этот процесс схематично представлен на рис. 30.

---

[73] В марте 2014 г. NTIA объявило о намерении передать ряд функций, связанных с управлением IANA, глобальному сообществу (см. http://ntia.doc.gov/press-release/2014/ntia-announces-intent-transition-key-internet-domain-name-functions). ICANN было поручено начать диалог со всеми заинтересованными группами для поиска модели системы управления IANA без участия NTIA. В результате этого процесса ICANN была создана дочерняя организация PTI (Public Technical Identifiers), выполняющая функции IANA. Об этом более подробно рассказано в разделе «Передача ключевых функций» в главе 4.

[74] https://www.icann.org/iana_imp_docs/129-root-zone-maintainer-service-agreement-v-28sep16

Рис. 30. Процесс внесения изменений в корневую зону.
Источник: Отчет об исследовании процесса внесения изменений в корневую зону (Root Zone Update Process Study, https://itp.cdn.icann.org/en/files/internet-assigned-numbers-authority-iana-functions/rzm-study-jas-icj-14-03-2022-14-03-2022-en.pdf)

Корневые серверы (КС)

Корневую зону обслуживают 13 DNS-серверов, также называемых корневыми. Имена КС начинаются с буквы латинского алфавита (от A до M) и имеют общее окончание root-servers.net, например, a.root-servers.net. Эта первая буква также используется как сокращенное имя сервера, например, сервер k.root-servers.net называют «K-сервер». Операторами КС являются различные организации, получившие право управления серверами в относительно отдаленном прошлом, когда подобные вопросы решались менее формально. Среди операторов — университеты, организации Минобороны США, некоммерческие ассоциации. За исключением оператора L-сервера, который находится под управлением ICANN, операторы финансово и юридически независимы от корпорации ICANN, в рамках которой действует IANA. При принятии операционных решений операторы руководствуются технической целесообразностью и существующими стандартами (например, RFC 2870[75]), в основном поддерживая статус-кво. Принято считать, что подобная независимость и разнородность операторов КС является основой технической и политической стабильности системы в целом, исключая узурпацию управления какой-либо из сторон.

Операторы КС образуют неформальную группу, цель которой — координация совместных действий и обмен операционной информацией и опытом. Группа

[75] RFC 2870: Root Name Server Operational Requirements, URL: https://www.rfc-editor.org/rfc/rfc2870

регулярно, три раза в год, проводит встречи, приуроченные к совещанию IETF. Одним из результатов таких совещаний является генерация секретного ключа KSK (Key Signing Key) для протокола TSIG. Члены группы также входят в Консультационный совет KSK ICANN (Root Server System Advisory Committee, RSSAC), среди задач которого — выработка рекомендаций по управлению KSK и внесению различных изменений в систему.

До недавнего времени отсутствовали какие-либо формальные отношения между операторами и ICANN/IANA. Эта ситуация изменилась с подписанием первого соглашения между ICANN и оператором F-сервера компанией ISC[76]. Данное соглашение не предусматривает никаких финансовых расчетов и лишь определяет взаимные обязанности сторон в отношении управления КС. Ряд операторов также обменялись письмами о взаимопонимании с ICANN (см., например, письмо от RIPE NCC[77]).

Ниже приведен список и краткая характеристика текущих операторов SKS (Synchronising Key Server – синхронизирующий сервер ключей)

Таблица 5. Список операторов SKS

КС	Организация-оператор	Характер деятельности, страна
A	VeriSign, Inc.	Коммерческая корпорация, один из крупнейших операторов DNS (например, .com, .net), поставщик средств защиты электронных коммуникаций, США
B	Information Sciences Institute	Институт Университета Южной Калифорнии (USC), США
C	Cogent Communications	Один из крупнейших коммерческих интернет-сервис-провайдеров, США
D	University of Maryland	Мэрилендский университет, США
E	NASA's Ames Research Center	Государственное агентство, США
F	Internet Systems Consortium, Inc.	Некоммерческая корпорация, США
G	U.S. DOD Network Information Center	Государственное агентство, США
H	U.S. Army Research Lab	Государственное учреждение, США
I	Netnod	Коммерческая организация, оператор точки обмена трафиком, оператор DNS, Швеция
J	VeriSign, Inc.	Коммерческая корпорация, один из крупнейших операторов DNS (например, .com, .net), поставщик средств защиты электронных коммуникаций, США
K	RIPE NCC	Некоммерческая ассоциация, РИР, Нидерланды
L	ICANN	Некоммерческая корпорация, США
M	WIDE Project	Некоммерческий проект, секретариат Университета Кейо, Япония

[76] https://www.icann.org/en/announcements/details/icann-board-approves-historic-f-root-agreement—first-formalization-of-mutual-responsibilities-between-isc-f-root-server-operator-and-icann-23-1-2008-en

[77] https://www.icann.org/en/system/files/files/pawlik-to-twomey-06may09-en.pdf

Альтернативные SKS

Так называемые альтернативные SKS появились из-за неудовлетворенности существующей системой управления SKS во главе с ICANN при участии правительства одной страны — США, а также по причине географической распределенности серверов и недостаточной поддержки интернационализированных доменов. Некоторые из таких SKS существуют до сих пор, например, Public-Root или Open Root Server Network (ORSN). Хотя эти системы копируют текущее состояние корневой зоны, сама архитектура предусматривает, что в определенных условиях альтернативные SKS могут предоставить альтернативное пространство имен. Администратор клиента DNS (обычно — сервера DNS, обслуживающего корпоративных пользователей или клиентов кабельной сети) может выбрать альтернативную SKS, просто изменив соответствующим образом файл hints.

Альтернативные SKS получили критическую оценку со стороны IETF как открывающие потенциальную возможность раскола единого Интернета (см. RFC 2826[78]).

Надо заметить, что масштабное внедрение аникаста в системе корневых серверов, а также поддержка ICANN интернационализированных имен существенно уменьшили необходимость в альтернативных серверах. Тем не менее, различные политические пертурбации, например, разоблачения Эдварда Сноудена, время от времени повышают активность групп, связанных с альтернативными SKS.

Экспериментальная СКС — Yeti DNS

Хотя за более чем 25-летнее существование в SKS произошел ряд существенных изменений — внедрение технологии аникаста, поддержка IPv6, подписание корневой зоны DNSSEC, — все они несли в себе риски, несмотря на значительную предварительную подготовку и тестирование. Причиной этому является то, что система SKS — единственная в своем роде, и лабораторное моделирование не способно отразить все многообразие экосистемы, в которой она существует. Взять хотя бы разнообразие DNS-клиентов или непредсказуемость различных промежуточных устройств — защитных сетевых экранов, балансировщиков нагрузки и т.п. Вследствие этого точно определить последствия того или иного изменения в системе SKS невозможно. Поскольку стабильность системы является наиболее важным требованием, возможность экспериментирования в SKS сводится к нулю. Это, в свою очередь, ведет к оссификации системы и затруднению ее дальнейшего развития.

В 2015 году несколько организаций (WIDE, BII и TISF) начали строительство экспериментальной SKS, получившей название Yeti DNS[79]. Эта система ис-

[78] RFC 2826: IAB Technical Comment on the Unique DNS Root, URL: https://www.rfc-editor.org/rfc/rfc2826

[79] https://yeti-dns.org

пользует точную копию официальной корневой зоны IANA (с точностью до замены адресов корневых серверов). Хотя по своей структуре она напоминает альтернативные SKS, задачей Yeti DNS является создание вовсе не альтернативного пространства имен, а параллельной системы для проведения экспериментов.

В частности, Yeti DNS может помочь ответить на следующие вопросы:

- Можно ли обеспечить работу SKS, используя исключительно IPv6?
- Каковы последствия более частой замены DNSSEC-ключей ZSK, например, каждые две недели?
- Каковы последствия более частой замены DNSSEC-ключей KSK, например, каждые шесть недель?
- Каково оптимальное число корневых серверов?
- Каковы последствия добавления или удаления оператора корневого сервера, насколько часто это можно делать?

Несколько опытов были успешно проведены, но проект испытывает ряд трудностей. Это связано, в первую очередь, с недостаточным объемом запросов, чтобы имитировать работу реальной SKS. Проблема усугубляется тем, что некоторые эксперименты могут отрицательно отразиться на качестве предоставляемой услуги. В результате некоторые участники Yeti DNS перестают использовать эту систему и переключаются на официальную SKS. Это, в свою очередь, еще более уменьшает нагрузку на Yeti DNS. В настоящее время проект продолжается, хотя и менее активно.

Локальный корень

Как мы уже видели, корневые серверы играют ключевую роль в процессе трансляции имен. Для каждого запроса к доменному имени для домена верхнего уровня, отсутствующего в кеше резолвера, этот запрос сначала отправляется к корневому серверу. Хотя время жизни ответов, полученных от корневых серверов, варьируется от одного до двух дней, отсутствие доступа к корневым серверам, например, вследствие атаки DDoS, приведет к отказу в обслуживании для всех доменов, чьи TLD отсутствуют в кеше резолвера. А если атака продлится достаточно долго, эта участь постепенно постигнет все запросы.

С другой стороны, подавляющая часть запросов к корневым серверам относится к несуществующим доменам, как следствие, значительные ресурсы системы тратятся впустую.

Документ «Запуск локального корневого сервера»[80] описывает подход, позволяющий решить обе проблемы.

[80] RFC 8806: Running a Root Server Local to a Resolver, URL: https://www.rfc-editor.org/rfc/rfc8806

Суть подхода заключается в том, что оператор рекурсивного резолвера имеет полную корневую зону локально, тем самым исключая необходимость внешних запросов к SKS. Основная идея состоит в том, чтобы создать службу постоянно обновляемой корневой зоны на том же хосте, что и резолвер, и использовать эту службу, когда резолверу требуется информация корневой зоны. Резолвер проверяет все ответы от корневой службы на том же хосте точно так же, как он проверяет все ответы от удаленного корневого сервера.

При этом необходимо учитывать несколько аспектов:

Во-первых, эта служба корневой зоны должна быть сконфигурирована таким образом, чтобы предотвратить доступ к ней какой-либо другой системы, кроме резолвера на том же хосте.

Во-вторых, содержимое корневой зоны должно обновляться с использованием таймеров из записи SOA в корневой зоне. По сути, это означает, что содержимое локальной корневой зоны, скорее всего, будет немного отставать от содержимого глобальных корневых серверов, поскольку эти серверы обновляются динамично, используя сообщения NOTIFY.

Корневую зону можно получить откуда угодно, если она содержит все записи DNSSEC, необходимые для проверки. В настоящее время можно получить корневую зону от ICANN путем передачи зоны AXFR[81] по протоколу TCP с DNS-серверов по адресам xfr.lax.dns.icann.org и xfr.cjr.dns.icann.org. Файл корневой зоны можно получить с помощью методов, описанных на странице Root Files[82]. В настоящее время корневая зона также может быть получена с помощью AXFR по TCP от следующих корневых серверов:

- b.root-servers.net
- c.root-servers.net
- d.root-servers.net
- f.root-servers.net
- g.root-servers.net
- k.root-servers.net

Здесь мы подходим к еще одной проблеме – каким образом удостовериться в целостности полученной локальной зоны. Простым ответом кажется использование DNSSEC. Однако не все записи зоны подписаны DNSSEC. Так, записи NS делегированных доменов и их IP-адреса не подписываются.

Другие способы защищенной передачи зоны, такие как TSIG, эфемерны и гарантируют лишь то, что клиент получит данные от ожидаемого сервера и что

[81] RFC 5936: DNS Zone Transfer Protocol (AXFR), URL: https://www.rfc-editor.org/rfc/rfc5936
[82] https://www.iana.org/domains/root/files

данные, отправленные сервером, не будут изменены во время передачи. Однако они не гарантируют, что сервер передает данные в первоначально опубликованном виде, и не предоставляют никаких методов для проверки данных, которые используются после завершения передачи.

Здесь на помощь приходит новая запись DNS – ZONEMD, – описанная в стандарте «Дайджест сообщений для зон DNS»[83]. Запись ZONEMD является криптографическим дайджестом данных в зоне. Это позволяет получателю зоны проверить целостность и подлинность зоны при использовании в сочетании с DNSSEC. ZONEMD является частью самой зоны, что позволяет проверять зону, независимо от того, каким способом она была получена.

Подписание корневой зоны

Как мы уже обсуждали, существенным недостатком базового протокола DNS является слабая система защиты данных. Ответы могут быть модифицированы «в полете» или путем создания ложных серверов. Их проникновение в кеши резолверов (так называемое отравление кеша) может производить продолжительный эффект. Радикальным решением проблемы является использование технологии DNSSEC, которая позволяет получателю ответа определить, были ли данные модифицированы.

В силу иерархического характера DNS подписание записей корневой зоны имело существенное значение как для усиления безопасности глобальной системы DNS, так и для более широкого внедрения DNSSEC.

Многие годы велись дискуссии о том, когда же будет подписана зона, кто будет контролировать ключи, как это отразится на системе DNS и Интернете в целом. После долгих публичных комментариев и внутренних дискуссий в 2009 году ICANN, VeriSign и Министерство торговли США договорились о прагматичной схеме, при которой существующий процесс внесения изменений в зону остается прежним, а основные игроки получают дополнительные роли: ICANN контролирует так называемый Trust Anchor — ключ для подписания ключей KSK, агентство Министерства торговли NTIA по-прежнему утверждает изменения, а VeriSign владеет ключом подписания зоны ZSK, который используется для генерирования подписанной корневой зоны, и осуществляет ее публикацию на скрытом мастер-сервере. Далее зона публикуется операторами корневой зоны. Надо отметить, что в соответствии с технологией DNSSEC ключ KSK подписывает ключ(и) ZSK. Другими словами, контроль за подписанием зоны в конечном итоге остается за ICANN.

Когда политические страсти вокруг подписания корневой зоны постепенно улеглись, настало время взглянуть на технические аспекты этого изменения. А их немало. Помимо внутренней защищенной архитектуры хранения и ис-

[83] RFC 8976: Message Digest for DNS Zones,
URL: https://www.rfc-editor.org/rfc/rfc8976

пользования ключей, а также защищенного взаимодействия между игроками, сама публикация подписанной зоны представляет серьезную задачу. Это, в первую очередь, связано с масштабом последствий, которые изменения в корневой зоне могут иметь для глобального сообщества пользователей Интернета. Специально созданная техническая группа, состоящая из экспертов в области DNS и безопасности, провела колоссальную работу по подготовке и воплощению данного масштабного проекта в жизнь.

Одной из основных задач, стоявших перед группой, являлось обеспечение постепенного внедрения технологии DNSSEC в корневой зоне. Было очевидно, что простая публикация подписанной зоны в один прекрасный момент была бы слишком рискованной и поэтому неприемлемой. Как поведут себя при этом разнообразные клиенты? Как это скажется на корневых серверах?

Как ни странно, основным изменением, связанным с внедрением DNSSEC в корневой зоне, являлось не собственно подписание зоны, а увеличение размера ответа на запрос клиента. Большие DNS-ответы подстерегают различные опасности: это и фрагментация пакетов на пути их следования, и невозможность их сборки клиентом, и фильтрация пакетов, превышающих по длине исторические 512 байт, маршрутизаторами и устройствами безопасности. Другими словами, при значительном увеличении размера ответов возрастает риск, что клиент не сможет получить ответ на запрос к корневому серверу.

Поэтому DNSSEC в корневой зоне было решено внедрять постепенно: сначала на одном сервере, потом на следующем и так далее, пока подписанная зона не будет публиковаться всеми 13 корневыми серверами (точнее, внедрение происходило по группам — всего шесть групп серверов). При этом велись наблюдения за возможным перераспределением нагрузки на серверы, чтобы исключить возможные осложнения. Ведь структурная «эмиграция» клиентов от сервера, на котором опубликована подписанная зона, скорее всего, означает, что клиенты либо не могут получить ответа от этого сервера, либо выполнение запроса занимает существенно больше времени, чем для остальных серверов.

Также следовало обеспечить возможность возвращения «на круги своя» — то есть к неподписанной корневой зоне — в случае каких-либо проблем. Было принято решение подписать зону таким образом, чтобы подписи не могли никоим образом подвергнуться проверке. Такую зону назвали DURZ — Deliberately Unvalidatable Root Zone, или корневая зона, которая не может быть криптографически проверена.

Процесс «распространения» DURZ по КС начался в январе 2010 года и занял почти полгода. 5 мая 2010 года DURZ была опубликована на последнем сервере — «J». Все прошло без каких-либо заметных негативных последствий в функционировании Интернета. Можно было переходить к следующему этапу — подписанию зоны «правильным» ключом.

16 июня семь доверенных представителей интернет-сообщества прибыли на площадку ICANN в местечке Калпепер (Culpeper) недалеко от Вашингтона для церемонии генерации ключей KSK и подписания ключей ZSK, которые в свою очередь используются для подписания записей самой корневой зоны.

15 июля 2010 года корневая зона была подписана этим ключом. Следуя передовой криптографической практике, ключи подписания зоны должны периодически обновляться. Новый пакет ключей ZSK генерируется четыре раза в год группой доверенных представителей интернет-сообщества. К концу 2016 года были проведены 27 таких церемоний, включая самую первую, в ходе которой был также сгенерирован первый ключ KSK. Однако и этот ключ необходимо периодически, хотя и гораздо реже, менять.

Однако замена ключа KSK является нетривиальной задачей. Дело в том, что в то время, как новые ключи ZSK (как и другие записи ресурсов) обновляются клиентом в ходе нормальных запросов DNS, ключ KSK, который является точкой доверия всей иерархии глобальной системы имен, конфигурируется администратором каждого резолвера вручную. До публикации стандарта RFC 5011[84] не существовало способа автоматического обновления KSK. Можно предположить, что не все резолверы поддерживают этот протокол, что делает задачу распределения нового KSK дополнительно сложной.

Для решения этой задачи группой экспертов был разработан план замены KSK, который был передан на обсуждение интернет-сообщества. План включал следующие шаги:

- Октябрь 2016: начало процесса замены KSK: генерация нового ключа KSK.
- Июль 2017: публикация нового KSK в DNS.
- Октябрь 2017: новый KSK используется для подписания пакета ключей корневой зоны, что, собственно, и означает замену ключа.
- Январь 2018: отзыв старого KSK.
- Март 2018: завершение процесса замены KSK.

Точка замены ключа (октябрь 2017 года в изначальном плане) является наиболее критическим моментом. Резолверы, осуществляющие проверку DNSSEC и не сконфигурировавшие новый ключ KSK (автоматически, используя протокол RC5011, или вручную администратором), выдадут ошибку валидации на любой запрос.

Эта проблема явилась причиной того, что замену ключа пришлось отсрочить на один год. Дело в том, что в начале 2017 года в IETF был разработан механизм, позволяющий резолверам, осуществляющим проверки DNSSEC, проинформи-

[84] RFC 5011: Automated Updates of DNS Security (DNSSEC) Trust Anchors, URL: https://www.rfc-editor.org/rfc/rfc5011

ровать DNS-сервер о том, какие ключи они используют для валидации[85]. Используя этот механизм, исследователи смогли получить более точное представление о проценте резолверов, готовых к замене ключа. К сожалению, как видно из рис. 31, на предполагаемый момент замены значительная часть резолверов (6-8%, это число выросло до 20% по мере увеличения числа измерений) по-прежнему доверяла старому ключу, а не новому. Этот процент начал уменьшаться только в мае 2018 года. На момент новой даты замены ключа – 11 октября 2018 года – только 5% наблюдаемых резолверов по-прежнему доверяли старому ключу, что являлось приемлемым для процесса замены.

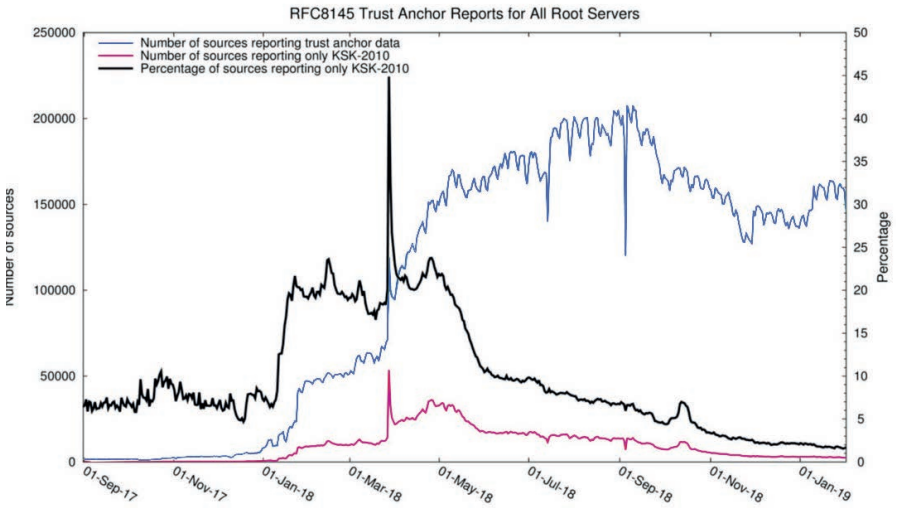

Рис. 31. Отчеты резолверов по использованию ключей KSK.
Источник: Обзор замены ключа DNSSEC KSK 2018 г.
https://www.icann.org/review-2018-dnssec-ksk-rollover.pdf

11 октября 2018 года произошла замена ключа. Ключи, используемые для подписания записей зоны, были подписаны новым ключом KSK. Этот момент также означал, что резолверы, которые не завершили переход, стали возвращать ошибку при попытке валидации DNSSEC. Тем не менее, процесс прошел без существенных инцидентов. По данным проекта Root Canary[86], в течение 48 часов после замены ключа 99% наблюдаемых резолверов использовали новый ключ[87].

[85] RFC 8145: Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC), URL: https://www.rfc-editor.org/rfc/rfc8145

[86] Проект был завершен в начале 2019 года, https://rootcanary.org/

[87] https://www.sidnlabs.nl/en/news-and-blogs/a-successful-root-ksk-rollover-a-short-look-back??language_id=2

Итак, замена ключа KSK прошла успешно и в будущем превратится в более или менее рутинную операцию. Гораздо более сложной является операция по замене криптографического алгоритма ключа. При первой замене ключа по-прежнему использовался широко распространенный алгоритм криптографической подписи RSA-SHA. В последнее время более распространенными стали новые криптографические алгоритмы. Это привело к обсуждению в сообществе ICANN необходимости подготовки процесса обновления будущих криптографических ключей DNS для реализации преимуществ этих новых алгоритмов.

В ноябре 2022 года ICANN пригласила добровольцев присоединиться к группе разработчиков плана изменения криптографического алгоритма, используемого для ключа подписи корневого ключа DNS и ключа подписи зоны. В конце февраля 2023 года такая группа была сформирована, о чем ICANN сообщила в своем анонсе[88].

Глобализация корневой зоны DNS. Программа gTLD

В 1984 году документ RFC 920 «Требования к доменам»[89] определил дополнительные домены верхнего уровня. К существовавшему домену .arpa, включавшему все хосты тогдашнего Интернета, были добавлены пять «смысловых» доменов: .gov для правительственных ресурсов, .edu для образовательных, .mil для военных, .com для коммерческих и .org для организаций. Разумеется, тогда все это относилось к американским организациям.

Данный документ также определил национальные домены и установил их формат — двухбуквенный код (alpha-2) таблицы ISO-3166.

Наконец, в документе была упомянута тогда еще пустая категория «мультиорганизаций» — больших транснациональных конгломератов, «не попадающих ни в одну из перечисленных категорий». До середины 1990-х годов корневая зона росла за счет национальных доменов, за исключением добавленных доменов .int и .net. Как заявил Джон Постел в 1994-м, «крайне маловероятно, что будут созданы новые домены верхнего уровня»[90]. Однако требования к созданию дополнительных доменов верхнего уровня значительно выросли, что привело к созданию нескольких групп и разработке соответствующих предложений.

Основной причиной этих требований явилась неудовлетворительная, с точки зрения мировой общественности, ситуация с так называемыми международными доменами. Именно таковыми со временем стали домены .com, .org и .net. Интернет переживал бум приватизации и стремительного развития, поэтому

[88] https://www.icann.org/ru/announcements/details/icann-convenes-design-team-to-evolve-root-zone-security-21-02-2023-ru

[89] RFC 920: Domain Requirements, URL: https://www.rfc-editor.org/rfc/rfc920

[90] RFC 1591: Domain Name System Structure and Delegation, URL: https://www.rfc-editor.org/rfc/rfc1591

регистрация имен и «международных» поддоменов стала носить глобальный характер. Но регистрация осуществлялась единственной центральной регистратурой Internic, обслуживаемой частной компанией Network Solutions (сегодня VeriSign). Проблема монополизации «международных» доменов требовала решения. Регулирование было невозможно, учитывая международный характер проблемы. Следовало привлекать рыночные механизмы. Решению вопроса «открытия» рынка корневой зоны и были посвящены новые предложения.

Одним из таких предложений явился так называемый Проект Постела[91], разработанный Джоном Постелом в 1996 году. Проект предусматривал создание нескольких комитетов, утверждающих образование новых доменов верхнего уровня. Решение носило технократический характер и предлагало осуществлять делегирование новых доменов верхнего уровня в том же стиле, в каком IANA назначает параметры протоколов. Проект Постела также предлагал передать управление IANA созданной в 1992 году организации ISOC (Internet Society) — «организационному дому» IETF. Этот проект вызвал критику со стороны общественности как ограничивающий конкуренцию.

Другое предложение было разработано группой под названием International Ad Hoc Committee (IAHC), специально созданной под эгидой ISOC, IAB, IANA, ITU, INTA и WIPO. Здесь была сделана попытка учесть недостатки проекта Постела и предложить более сбалансированную модель управления. Кстати, именно IAHC предложил использование термина «общие домены верхнего уровня» (gTLD, generic TLD) вместо «международных» (iTLD, international TLD), использовавшихся ранее. Комитет прекратил свое существование в мае 1997 года после публикации предложения. Хотя рекомендации этой группы не были воплощены в жизнь, многие из них легли в основу деятельности созданной вскоре корпорации ICANN[92].

Прошло чуть меньше двух лет после создания ICANN, и уже в 2000 году было анонсировано создание семи новых имен: .aero, .biz, .coop, .info, .museum, .name, .pro, которые постепенно появились в корневой зоне к 2003 году.

Следующий этап расширения корневой зоны прошел под флагом так называемых спонсированных доменов. Спонсорами этих имен являлись этнические, профессиональные или географические сообщества.

Наконец, в 2005 году gNSO начала рассмотрение вопроса о более масштабном создании общих доменов верхнего уровня. Эта работа базировалась на опыте внедрения ICANN доменов в 2000–2003 годах, а процесс разработки соответствующих политик занял два года. В результате были утверждены 19 рекомен-

[91] https://datatracker.ietf.org/doc/draft-postel-iana-itld-admin/

[92] С текстом предложения можно ознакомиться на https://web.archive.org/web/20070426031850/http://www.gtld-mou.org:80/draft-iahc-gTLDspec-00.html

даций по созданию новых gTLD, включающих критерии выбора имени и договорные условия. Программа создания новых gTLD вызвала немало критики со стороны общественности. Практические ответы на многие вопросы — о защите прав правообладателей, финансовых последствиях регистрации тех или иных имен в качестве доменов верхнего уровня и т.п. — нам еще предстоит увидеть. Программа также вызвала обеспокоенность технического сообщества. Для ответа на наиболее острые вопросы, связанные с устойчивостью и масштабируемостью корневой зоны и SKS, было предпринято несколько исследований как самой корпорацией ICANN, так и комитетами внутри корпорации — SSAC и RSSAC[93]. Результаты этой работы, разумеется, содержат элемент неопределенности. Но стало очевидным, что расширение корневой зоны примерно на 1000 новых доменов в год хотя и требует периодической переоценки и мониторинга, на данном этапе не представляет существенного технического риска.

В июне 2011 года совет директоров ICANN объявил запуск программы новых gTLD. Прием заявок на них был начат 12 января 2012 года. В итоге было получено 1930 заявок. На июнь 2014 года 272 заявки были реализованы в виде новых доменов gTLD, таких как .website, .vodka, .club, .luxury, .москва, .дети и т.д. Спустя два года число доменов верхнего уровня выросло более чем в четыре раза, достигнув почти 1200 доменов верхнего уровня. Однако после первоначального ажиотажа рост существенно замедлился, и с тех пор добавилось только несколько десятков доменов.

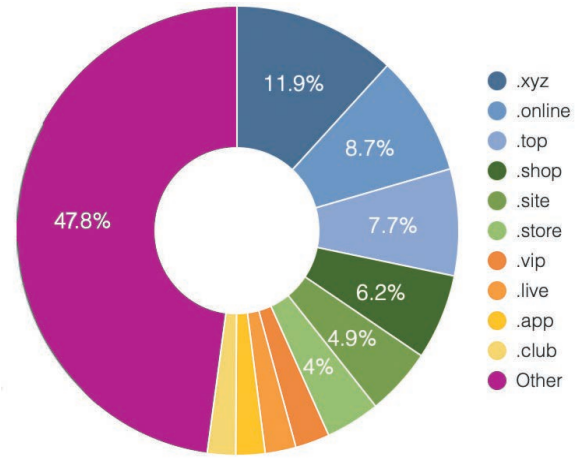

Рис. 32. Доля регистраций доменов второго уровня в new gTLD (июль 2023).
Источник: https://ntldstats.com/tld

[93] https://www.icann.org/en/announcements/details/new-gtlds-root-zone-scaling-report-27-6-2012-en

Очевидно, что емкость рынка ограничена, и больше половины всех доменов второго уровня зарегистрированы в десяти крупнейших gTLD.

Однако на повестке дня стоит подготовка к следующему раунду программы new gTLD. Этому предшествовала значительная работа по подведению итогов раунда 2012 года и выработка рекомендаций по улучшению.

Уже в конце 2015 года Совет gNSO инициировал процесс разработки политики и создал рабочую группу по последующим процедурам для новых gTLD. Перед рабочей группой была поставлена задача использовать коллективный опыт сообщества, полученный в ходе первого раунда программы New gTLD 2012 года, чтобы определить, какие изменения, если таковые имеются, необходимо внести в существующие рекомендации по политике введения новых родовых доменов верхнего уровня от 8 августа 2007. Эта работа была завершена 18 февраля 2021 года утверждением и публикацией «Итогового отчета о процессе разработки политики последующих процедур для программы new gTLD»[94].

Следующим шагом является работа по реализации рекомендаций, содержащихся в итоговом отчете, в процессе разработки политики последующих процедур для новых gTLD. Результатом этого процесса станет обновленное «Руководство кандидата» (Applicant Guidebook, AGB). Также ICANN работает над планом реализации нового раунда.

Учитывая все эти задачи, корпорация ICANN ожидает, что AGB будет завершено во втором квартале 2025 года, что позволит начать прием заявок во втором квартале 2026.

Заключение

Глобальная система доменных имен DNS имеет интересное свойство, противоположное сетевой самоорганизующейся архитектуре Интернета. DNS и в архитектурном и, что более важно, в операционном смыслах — иерархическая, хотя и распределенная система. В DNS существует только один корень, и проблемы на этом уровне затрагивают всю систему.

Тем не менее, эта система успешно идет в ногу с самим Интернетом без фундаментальных изменений во внутренних протоколах и архитектуре. Более того, в качестве глобальной распределенной базы данных DNS уже давно не ограничивается исполнением той простой функции, для которой

[94] Final Report on the new gTLD Subsequent Procedures Policy Development Process, https://gnso.icann.org/sites/default/files/file/field-file-attach/final-report-newgtld-subsequent-procedures-pdp-20jan21-en.pdf

была создана, — трансляции имен в адреса IP. Система DNS активно используется во многих новациях: как в хороших — например, при оптимизации доставки контента, так и в плохих — например, в целях управления ботнетами в руках злоумышленников.

Еще одной особенностью DNS является то, что эта система всегда на виду — она наиболее понятна пользователю Интернета. Мы набираем доменное имя и получаем доступ к веб-серверу или другому ресурсу. Не случайно вопросы интернационализации DNS довольно остро стояли и отчасти продолжают стоять на повестке дня. Проблематика DNS также широко обсуждается и на политическом уровне — в рамках дискуссий об управлении Интернетом.

Другие аспекты DNS, возможно, менее заметны, но успешное решение связанных с ними вопросов не менее важно. В первую очередь имеются в виду вопросы безопасности. Хотя спецификации расширений безопасности DNS — DNSSEC — были стандартизованы IETF еще в 2005 году, уровень их внедрения не внушает большого оптимизма. А ведь DNSSEC может открыть новые возможности использования DNS — в частности, с применением сертификатов TLS веб- и других ресурсов в соответствии со спецификациями DANE.

И пусть иногда кажется, что поисковые машины постепенно уменьшают значимость DNS, заменяя адреса ресурсов Сети на ключевые слова — объекты запросов, но в реальности DNS вряд ли скоро уступит свои позиции.

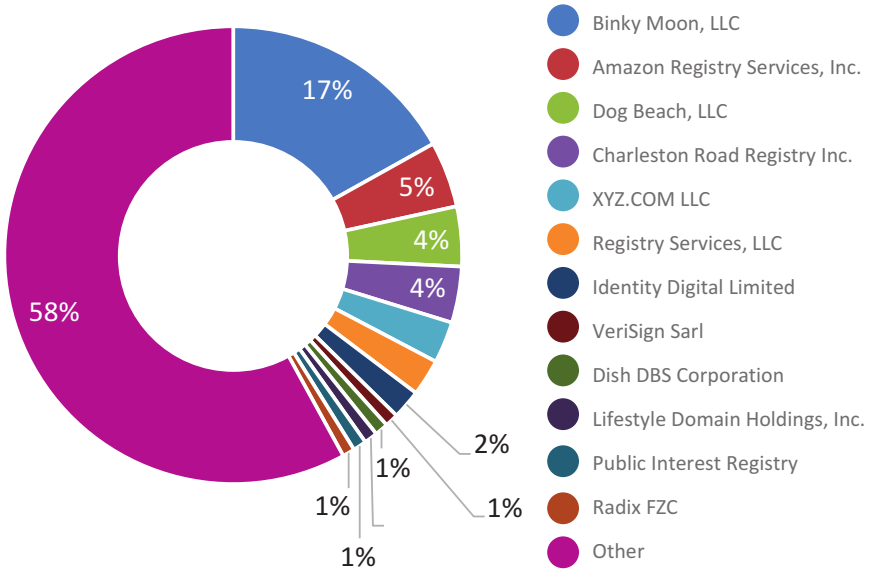

Рис. 33. Распределение рынка доменов new gTLD между операторами регистратур (состояние март 2023 г.).
Источник: www.icann.org/en/system/files/files/rr-voting-status-xls-20mar23-en.xlsx

Очевидно, что емкость рынка ограничена, и больше половины всех доменов второго уровня зарегистрированы в десяти крупнейших gTLD, как видно из рисунка 32.

И в целом рынок new gTLD является весьма сконцентрированным. Дюжина операторов регистратур владеют 42% всех доменов new gTLD, см. рис 33.

Подобная ситуация наблюдается и на рынке регистраторов. Почти 60% рынка всех зарегистрированных доменов второго уровня принадлежит 10 компаниям (рис 34).

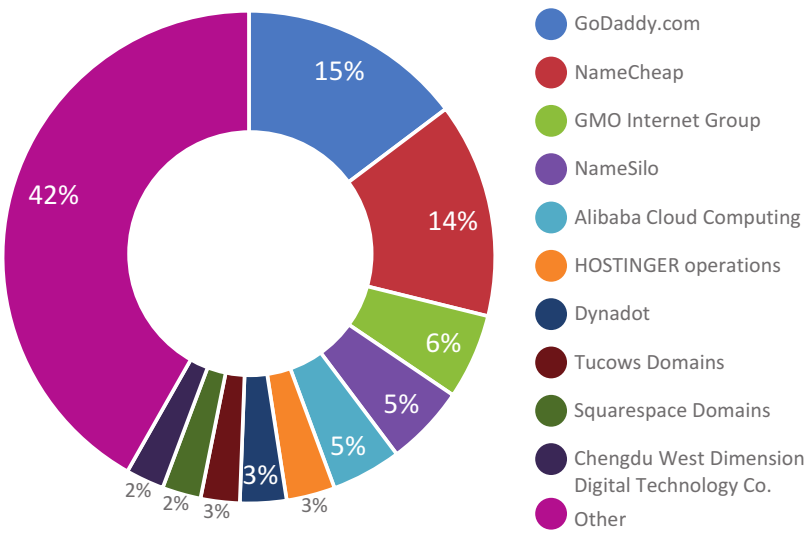

Рис. 34. Распределение рынка new gTLD между регистраторами по числу доменов второго уровня (состояние февраль 2024 г.).
Источник: https://ntldstats.com/registra