Internet Engineering Task Force (IETF)
Request for Comments: 9460
Category: Standards Track
ISSN: 2070-1721

B. Schwartz
Meta Platforms, Inc.
M. Bishop
E. Nygren
Akamai Technologies

November 2023

Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records)

Привязка служб и спецификация параметров через DNS (записи о ресурсах SVCB и HTTPS)

Аннотация

Этот документ задаёт типы записей о ресурсах (RR) DNS SVCB (Service Binding - привязка сервиса) и HTTPS для облегчения поиска сведений, требуемых для организации соединений с сетевыми службами, такими как источники HTTP. Записи SVCB позволяют предоставлять услугу в нескольких дополнительных точках, с каждой из которых связаны параметры (такие как настройка транспортного протокола), и являются расширяемыми для поддержки будущих применений (таких как ключи шифрования TLS ClientHello). Кроме того, они разрешают псевдонимы вершин доменов, которые невозможны с CNAME. HTTPS RR является вариантом SVCB для применения с HTTP (см. RFC 9110). Предоставляя клиенту больше информации до его попытки организовать соединение, эти записи обеспечивают преимущества в плане производительности и приватности.

Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF[1] и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG[2]. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке https://www.rfc-editor.org/info/rfc9460.

Оглавление

1. Введение....2
1.1. Цели....3
1.2. Обзор SVCB RR....3
1.3. Терминология....3
2. Тип записи SVCB....4
2.1. Формат представления в файле зоны....4
2.2. Формат передачи RDATA....4
2.3. Имена в запросах SVCB....5
2.4. Интерпретация....5
2.4.1. SvcPriority....5
2.4.2. AliasMode....5
2.4.3. ServiceMode....6
2.5. Специальная обработка . в TargetName....6
2.5.1. AliasMode....6
2.5.2. ServiceMode....6
3. Поведение клиента....6
3.1. Обработка отказов при распознавании....7
3.2. Клиенты, использующие прокси....7
4. Поведение серверов DNS....7
4.1. Полномочный сервер....7
4.2. Рекурсивный распознаватель....7
4.2.1. DNS64....8
4.3. Общие требования....8
4.4.Подсеть клиента EDNS....8
5. Оптимизация производительности....8

[1]Internet Engineering Task Force - комиссия по решению инженерных задач Internet.
[2]Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

5.1. Оптимистическое соединение заранее и повторное использование....8
5.2. Генерация и использование неполных откликов....9
6. Совместимые с SVCB типы RR....9
7. Исходные SvcParamKey....9
7.1. alpn и no-default-alpn....9
7.1.1. Представление....9
7.1.2. Использование....10
7.2. port....10
7.3. ipv4hint и ipv6hint....10
7.4. mandatory....11
8. Совместимость ServiceMode RR и обязательные ключи....11
9. Использование привязки сервиса к HTTP....11
9.1. Имена в запросах HTTPS RR....11
9.2. Сравнение с Alt-Svc....11
9.2.1. Использование ALPN....12
9.2.2. Недоверенные каналы....12
9.2.3. Срок действия кэша....12
9.2.4. Детализация....12
9.3. Взаимодействие с Alt-Svc....12
9.4. Требование указания имени сервера....12
9.5. Строгая транспортная защита HTTP (HSTS)....12
9.6. Использование HTTPS RR в других протоколах....13
10. Структура зоны....13
10.1. Структурирование зон для гибкости....13
10.2. Структурирование зон для производительности....13
10.3. Вопросы эксплуатации....13
10.4. Примеры....14
10.4.1. Улучшения протокола....14
10.4.2. Псевдонимы на вершине зоны....14
10.4.3. Привязка параметров....14
10.4.4. Конфигурация с несколькими CDN....14
10.4.5. Использование с другими протоколами....15
11. Взаимодействие с другими стандартами....15
12. Вопросы безопасности....15
13. Вопросы приватности....16
14. Взаимодействие с IANA....16
14.1. Тип SVCB RR....16
14.2. Тип HTTPS RR....16
14.3. Новый реестр для параметров сервиса....16
14.3.1. Процедура....16
14.3.2. Исходное содержимое....16
14.4. Другие обновления реестров....17
15. Литература....17
15.1. Нормативные документы....17
15.2. Дополнительная литература....18
Приложение A. Декодирование текста в файлах зон....18
A.1. Декодирование списка через запятые....19
Приложение B. Сводка отображений HTTP....19
Приложение C. Сравнение с другими вариантами....19
C.1. Отличия от типа SRV RR....19
C.2. Отличия от предложенной записи HTTP....19
C.3. Отличия от предложенной записи ANAME....20
C.4. Сравнение с раздельными типами RR для AliasMode и ServiceMode....20
Приложение D. Тестовые векторы....20
D.1. AliasMode....20
D.2. ServiceMode....20
D.3. Отказы....22
Благодарности и предложения....22
Адреса авторов....23

1. Введение

Записи о ресурсах (resource record RR) SVCB (Service Binding - привязка сервиса) и HTTPS дают клиентам полные инструкции для доступа к сервису. Это позволяет повысить производительность и приватность за счёт исключения временных соединений и неоптимальным сервером, заданным по умолчанию, согласования предпочтительного протокола и предоставления соответствующих открытых ключей. Например, в настоящее время клиенты HTTP распознают лишь записи A и AAAA для имени хоста-источника, узнавая лишь адреса IP. Если клиент HTTP узнает больше об источнике до соединения с ним, он может сменить http на https, разрешить HTTP/3 или Encrypted ClientHello [ECH] или переключиться на более предпочтительную конечную точку. Очень желательно минимизировать число поисков и круговых обходов, требуемых для получения этой дополнительной информации.

Записи SVCB и HTTPS также помогают передаче оператором операционных полномочий одному или нескольким другим доменам, например, путём создания псевдонима https://example.com для конечной точки svc.example.net. Хотя это можно реализовать с помощью CNAME, такое решение подходит не всегда. CNAME не подходит, если оператору сервиса нужно предоставлять связанную коллекцию согласованных параметров конфигурации (например, местоположение в сети, протокол, ключи) через DNS.

В этом документе описываются SVCB RR как записи общего назначения, которые эффективно и напрямую применимы к широкому спектру служб (раздел 2). Описаны также правила определения других типов RR, совместимых с SVCB

(раздел 6), начиная с типа HTTPS RR (раздел 9), обеспечивающего эффективную и удобную работу с HTTP за счёт исключения потребности в метек Attrleaf [Attrleaf] (параграф 9.1).

Для SVCB RR имеется два режима - 1) AliasMode, где просто передаётся операционное управление ресурсом, и 2) ServiceMode, где собираются воедино сведения о конфигурации конечной точки сервиса. В ServiceMode обеспечиваются дополнительные параметры key=value для каждого набора RDATA.

1.1. Цели

Цель SVCB RR - позволить клиентам распознавать одну дополнительную запись DNS RR, как указано ниже.

- Предоставить дополнительные конечные точки, полномочные для сервиса, вместе с параметрами, связанными с каждой из них.
- Не предполагать, что все конечные точки имеют одинаковые параметры или даже управляются одним органом. Это важно, поскольку DNS не предоставляет способа связать вместе несколько RRset для одного имени. Например, если www.example.com является псевдонимом CNAME, который переключается между тремя сетями доставки содержимого (Content Delivery Network или CDN) или средами хостинга, последовательные запросы для этого имени могут возвращать записи, соответствующие разным средам.
- Обеспечить похожую на CNAME функциональность на вершине зоны (например, example.com) для участвующих протоколов и, как правило, распространить операционные полномочия для сервиса, указанного доменным именем, на другие экземпляры с иными именами.

Дополнительные цели, относящиеся к HTTPS RR и вариантам использования HTTP, включают:

- прямое подключение к дополнительным конечным точкам HTTP/3 (транспорт QUIC) [HTTP/3];
- поддержка отличных от принятых по умолчанию портов TCP и UDP;
- обеспечение похожих на SRV преимуществ (например, псевдонимы на вершине зоны, как отмечено выше) для HTTP, где записи SRV [SRV] не получили широкого распространения;
- предоставление указаний, сообщающих, что следует применять схему https вместо http во всех запросах HTTP для данного хоста и порта, подобно строгой транспортной защите HTTP (Strict Transport Security) [HSTS] (см. параграф 9.5);
- возможность передачи зашифрованных ключей приветствия (Encrypted ClientHello) [ECH] связанных с дополнительной конечной точкой.

1.2. Обзор SVCB RR

В этом параграфе рассматривается запись SVCB RR с прямыми ссылками на описание каждого компонента (как указано в разделе 6, это применимо и к HTTPS RR с тем же кодированием, форматом и семантикой высокого уровня).

SVCB RR имеет два режима - 1) AliasMode (параграф 2.4.2), где псевдоним передаётся друнрму имени, и 2) ServiceMode (параграф 2.4.3), где предоставляются сведения о соединении, привязанные к домену конечной точки сервиса. Два типа записи RR позволяют клиентам получать требуемые сведения в одном запросе (параграф 2.3).

SVCB RR имеет два обязательных поля и одно необязательное.

SvcPriority (параграф 2.4.1)
: Приоритет этой записи (относительный с предпочтением меньших значений). Значение 0 указывает AliasMode.

TargetName
: Доменное имя целевого псевдонима (AliasMode) или дополнительной конечной точки (ServiceMode).

SvcParams (необязательно)
: Список пар key=value, описывающих дополнительную конечную точку в TargetName (используется только в ServiceMode, иначе игнорируется). Описание SvcParams приведено в параграфе 2.1.

Сотрудничающие рекурсивные распознаватели DNS выполняют последовательное распознавание записей (SVCB, A, AAAA) и возвращает результат в разделе отклика Additional (параграф 4.2). Клиенты используют записи из раздела Additional, возвращаемые рекурсивным распознавателем, или выполняют требуемое распознавание записей SVCB, A, AAAA (раздел 3). Полномочные серверы DNS добавлять записи SVCB, A, AAAA, CNAME в раздел Additional откликов на запрос SVCB (параграф 4.1).

В ServiceMode параметры SvcParams записи SVCB RR обеспечивают расширяемую модель данных для описания дополнительных конечных точек, которые полномочны для сервиса, вместе со связанными с каждой из этих точек параметрами (раздел 7).

Для случаев использования HTTP запись HTTPS RR (раздел 9) обеспечивает многие преимущества Alt-Svc [AltSvc] без ожидании полного инициирования соединения HTTP (несколько круговых обходов) до выяснения предпочтительного варианта и необходимости раскрытия предполагаемой цели пользователя всем элементам на пути через сеть.

1.3. Терминология

Терминология в этом документе основана на общем случае применения записи SVCB для доступа к ресурсу, указанному URI, где поле содержит имя хоста DNS как указание хоста.

- Служба (service) - источник информации, указанный полями authority и scheme в URI и способный предоставить доступ к ресурсу. Для https URI service соответствует origin [RFC6454].
- Имя службы (service name) - хостовая часть authority.
- Полномочная конечная точка (authority endpoint) - имя полномочного хоста и номер порта, подразумеваемый схемой или указанный в URI.

- Дополнительная конечная точка (alternative endpoint) - имя хоста, номер порта и связанные с ними инструкции для клиента о доступе к экземпляру сервиса.

Дополнительная терминология DNS соответствует [DNSTerm].

SVCB - это сокращение от service binding (привязка сервиса). SVCB RR, HTTPS RR и будущие типы RR, использующие форматы и реестр SVCB, называются совместимыми с SVCB типами RR. Для обозначения этой системы в целом также применяется сокращение SVCB.

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с BCP 14 [RFC2119] [RFC8174] тогда и только тогда, когда они выделены шрифтом, как показано здесь.

2. Тип записи SVCB

Тип SVCB DNS RR (64) служит для указания дополнительных конечных точек сервиса. Алгоритм распознавания записей SVCB и связанных с ними адресных записей задан в разделе 3.

По мере необходимости могут определяться другие совместимые с SVCB типы RR (см. раздел 6). В частности, HTTPS RR (тип 65) обеспечивает специальную обработку для источников https, как описано в разделе 9.

Записи SVCB RR могут расширяться списками SvcParams, содержащими пары SvcParamKey и SvcParamValue. Для каждого SvcParamKey имеется имя представления и зарегистрированный номер. Формат значения зависит от SvcParamKey. Для каждого SvcParam имеется конкретный формат представления (в файлах зон) и кодирование в линии (например, доменные имена, двоичные данные или численные значения). Исходные SvcParamKey и их форматы определены в разделе 7.

2.1. Формат представления в файле зоны

Формат представления <RDATA> записи ([RFC1035], параграф 5.1) имеет вид

```
SvcPriority TargetName SvcParams
```

Записи SVCB определены специально для класса Internet (IN) ([RFC1035], параграф 3.2.4). SvcPriority - это число из диапазона 0-65535, TargetName - <domain-name> ([RFC1035], параграф 5.1), а SvcParams - список разделённых пробелами SvcParam в виде SvcParamKey=SvcParamValue или просто SvcParamKey. SvcParamKey регистрируются IANA (параграф 14.3).

Каждый ключ SvcParamKey **нужно** указывать в SvcParams не более 1 разе. В формате представления SvcParamKey указываются алфавитно-цифровыми строками в нижнем регистре. Имена ключей содержат от 1 до 63 символов из диапазонов a-z, 0-9 и -. В ABNF [RFC5234] это имеет вид

```
alpha-lc      = %x61-7A   ; a-z
SvcParamKey   = 1*63(alpha-lc / DIGIT / "-")
SvcParam      = SvcParamKey ["=" SvcParamValue]
SvcParamValue = char-string ; See Appendix A.
value         = *OCTET ; значение до зависящего от ключа разбора
```

Значения SvcParamValue разбираются с использованием алгоритма декоирования строк (Приложение A), дающего значение, которое затем проверяется и преобразуется в формат передачи, зависящий от ключа. Если необязательные = и SvcParamValue отсутствуют, значение считается пустым.

Произвольные ключи могут быть укзаны с использованием формата представления неизвестных ключей keyNNNNN, где NNNNN - численное значение ключа без нулей в начале. SvcParam в этой форме **нужно** разбирать, как указано выше, а декодированное значение **нужно** использовать как кодирование для передачи.

Для некоторых SvcParamKey значение соответствует списку или набору элементов. В формате представления для таких ключей **следует** использовать список через запятые (Приложение A.1).

SvcParam в формате представления **можно** указывать в любом порядке, но повторение ключей **недопустимо**.

2.2. Формат передачи RDATA

RDATA для записи SVCB RR состоит из:

- 2-октетного поля SvcPriority в форме целого числа с сетевым порядком байтов;
- несжатого полного имени TargetName в форме меток в соответствии с параграфом 3.1 в [RFC1035];
- SvcParams, составляющие остальную часть записи (менее 65535 октетов с учётом ограничений на размер RDATA и сообщения DNS).

При непустом списке SvcParams он содержит последовательность пар SvcParamKey=SvcParamValue в форме:

- 2-октетного номера SvcParamKey в сетевом порядке байтов (значения приведены в параграфе 14.3.2);
- 2-октетного поля размера SvcParamValue в форме целого числа от 0 до 65535 с сетевым порядком байтов;
- строки октетов указанного размера со значением SvcParamValue в формате, определяемом SvcParamKey.

SvcParamKey **нужно** указывать в порядке роста номеров.

Клиенты **должен** считать недействительной RR:

- с завершением RDATA внутри SvcParam;
- с размещением SvcParamKeys не в порядке роста номеров;
- с SvcParamValue для SvcParamKey в неожиданном формате.

Отметим, что второе условие предполагает отсутствие дубликатов SvcParamKey.

Если какая-либо из RR имеет некоррктную форму, клиент **должен** отвергнуть RRset целиком и вернуться к организации соединения без SVCB.

2.3. Имена в запросах SVCB

При запросе SVCB RR сервис транслируется в QNAME путём добавления перед именем службы метки, указывающей схему, с префиксом _, что создаёт доменное имя типа _examplescheme.api.example.com.. Это соответствует шаблону именования Attrleaf [Attrleaf], поэтому схема **должна** регистрироваться в IANA (см. раздел 11).

Документы по отображению протоколов **могут** задавать другие метки с префиксом _, добавляемые в начало. Для схем, указывающих порт (параграф 3.2.3 в [URI]), одним из разумных вариантов является указание в начале номера порта, если он отличается от принятого по умолчанию. В данном документе это называется именованием с префиксом порта (Port Prefix Naming) и применяется в примерах.

Сведения о поведении HTTPS RR приведены в параграфе 9.1.

Когда предшествующая запись CNAME или SVCB имеет псевдонимом запись SVCB, каждую RR **нужно** возвращать со своим именем владельца, как при обычной обработке CNAME ([RFC1034], параграф 3.6.2). Более подробные сведения приведены в рекомендациях по работе с псевдонимами для клиентов (раздел 3), серверов (раздел 4) и зон (раздел 10).

Отметим, что ни одна из этих схем не меняет источник или полномочия для целей проверки. Например, клиенты TLS **должны** проверять сертификаты TLS для исходного имени сервера.

Например, владелец example.com может опубликовать запись

```
_8443._foo.api.example.com. 7200 IN SVCB 0 svc4.example.net.
```

Эта запись указывает, что foo://api.example.com:8443 является псевдонимом для svc4.example.net. Владелец example.net, в свою очередь, может опубликовать запись

```
svc4.example.net.  7200  IN SVCB 3 svc4.example.net. (
    alpn="bar" port="8004" )[1]
```

Данная запись будет указывать, что эти службы поддерживаются на порту с номером 8004, поддерживающем протокол bar и связанный с ним транспорт, дополняющий принятый по умолчанию протокол для foo://.

2.4. Интерпретация

2.4.1. SvcPriority

При SvcPriority = 0 запись SVCB указывает AliasMode (параграф 2.4.2), иначе - ServiceMode (параграф 2.4.3).

Внутри SVCB RRset всем RR **следует** иметь один режим. Если RRset содержит запись в AliasMode, получатель **должен** игнорировать все записи ServiceMode в наборе.

Наборы RRset являются явно неупорядоченными, поэтому для упорядочения SVCB RR применяется поле SvcPriority. Меньшее значение SvcPriority указывает, что владелец домена рекомендует использовать эту запись, а не ServiceMode RR с большим значением SvcPriority. При получении RRset с несколькими записями SVCB с одинаковым значением SvcPriority клиенту **следует** делать случайный выбор среди них для равномерного распределения нагрузки.

2.4.2. AliasMode

В режиме AliasMode запись SVCB назначает службе псевдоним TargetName. Наборам SVCB RRset **следует** иметь лишь одну RR в режиме AliasMode, а при наличии нескольких AliasMode RR клиенту или рекурсивному распознавателю **следует** выбирать одну из них случайным образом.

Основным назначением AliasMode является поддержка псевдонимов на вершине зоны, где записи CNAME не разрешены (см., например, параграф 2.4 в [RFC1912]). В AliasMode TargetName будет именем домена, которое распознаётся в записи SVCB, AAAA и/или A (псевдонимы для типов RR, совместимых с SVCB, рассмотрены в разделе 6). В отличие от CNAME, записи AliasMode не влияют на распознавание других типов RR и применяются лишь к конкретному сервису, а не ко всему доменному имени.

TargetName в AliasMode **не следует** совпадать с именем владельца, поскольку это создаст петлю. В AliasMode получатели **должны** игнорировать имеющиеся SvcParams. Анализатор файлов зон **может** выдавать предупреждение, если запись AliasMode включает SvcParams. Использование SvcParams в AliasMode в настоящее время не определено, но будущие спецификации могут расширить записи AliasMode для включения SvcParams.

Оператор foo://example.com:8080 может направить запрос к службе, работающей на foosvc.example.net, записью

```
_8080._foo.example.com. 3600 IN SVCB 0 foosvc.example.net.
```

Использование AliasMode позволяет разделить задачи - владелец foosvc.example.net может добавлять и удалять записи SVCB ServiceMode без необходимости соответствующего изменения example.com. Отметим, что при обещании foosvc.example.net всегда публиковать запись SVCB можно заменить AliasMode записью CNAME с тем же именем владельца.

Режим AliasMode особенно полезен для совместимых с SVCB типов RR, которым не требуется префикс с подчёркиванием, таких как HTTPS RR. Например, оператор https://example.com может направлять запросы серверу svc.example.net, публикуя на вершине зоны запись

```
example.com. 3600 IN HTTPS 0 svc.example.net.
```

Отметим, что имя владельца записи SVCB **может** быть каноническим именем записи CNAME, а TargetName **может** быть владельцем CNAME. Клиенты и рекурсивные распознаватели **должны** следовать записям CNAME как обычно. Для предотвращения неограниченных цепочек псевдонимов клиенты и рекурсивные распознаватели **должны** задавать предел числа псевдонимов SVCB, используемых для каждого запроса распознавания. **Недопустимо** устанавливать

[1] Здесь и далее скобки () в примерах используются для переноса длинных строк файлов зон.

для предела значение 0, т. е. реализации **должны** быть способны следовать хотя бы одной записи AliasMode. Конкретное значение предела задаёт реализация. Зоны, требующие следовать нескольким записям AliasMode могут столкнуться с проблемами совместимости и производительности. Поскольку устаревшие клиенты не будут знать, как использовать эти записи, операторам сервиса, вероятно, придётся сохранять записи AAAA и A для отката вместе с записью SVCB, хотя в общем случае запись SVCB может обеспечивать более высокую производительность и поэтому будет предпочтительней для клиентов, реализующих эту спецификацию.

Записи AliasMode применимы лишь для запросов конкретного типа RR. Например, запись SVCB не может быть псевдонимом для HTTPS и наоборот.

2.4.3. ServiceMode

В режиме ServiceMode элементы TargetName и SvcParams в каждой RR связывают с каждой дополнительной конечной точкой сервиса параметры соединения с нею. Каждая схема протокола, использующая SVCB, **должна** задать отображение протокола, разъясняющее использование SvcParams для соединений по этой схеме. Если в отображении не указано иное, клиенты **должны** игнорировать все нераспознанные SvcParam.

Некоторые SvcParams задают требования к другим SvcParams в RR. ServiceMode RR называют самосогласованной, если её SvcParams соответствуют требованиям друг друга. Клиенты **должны** отклонять RR, где распознанные SvcParams не согласованы, и **могут** отвергать RRset целиком. Чтобы помочь операторам в предотвращении таких ситуаций реализациям файлов зон также **следует** обеспечивать самосогласованность.

2.5. Специальная обработка . в TargetName

Если TargetName имеет значение . (метка нулевого размера в формате передачи), применяются специальные правила.

2.5.1. AliasMode

В SVCB RR AliasMode элемент TargetName со значением . указывает, что сервис недоступен или не существует. Это рекомендация для клиентов, которую они **могут** игнорировать и пытаться соединиться без применения SVCB.

2.5.2. ServiceMode

В SVCB RR ServiceMode элемент TargetName со значением . говорит, что имя владельца этой записи **должно** использоваться в качестве действующего TargetName. Если запись имеет шаблонное имя владельца в файле зоны, получателю **нужно** использовать синтезированное имя владельца в качестве действующего TargetName. В приведённом ниже примере svc2.example.net является действующим значением TargetName.

```
example.com.      7200  IN HTTPS 0 svc.example.net.
svc.example.net.  7200  IN CNAME svc2.example.net.
svc2.example.net. 7200  IN HTTPS 1 . port=8002
svc2.example.net. 300   IN A     192.0.2.2
svc2.example.net. 300   IN AAAA  2001:db8::2
```

3. Поведение клиента

Распознавание SVCB - это процесс перечисления и упорядочивания доступных конечных точек сервиса, выполняемый клиентом, как указано ниже.

1. Пусть $QNAME - имя службы с соответствующими префиксами для схемы (см. параграф 2.3).
2. Выдается запрос SVCB для $QNAME.
3. Если для $QNAME возвращается запись SVCB AliasMode (после обычного следования CNAME), для $QNAME устанавливается устанавливается значение TargetName (без дополнительных префиксов) с возвратом к п. 2 с учётом ограничения размера и эвристики обнаружения петель (см. параграф 3.1).
4. При возврате обной или нескольких «совместимых» (раздел 8) записей ServiceMode они представляют дополнительные конечные точки. Записи сортируются по возрастанию SvcPriority.
5. Иное говорит о неудаче распознавания SVCB и пустом списк дополнительных конечных точек.

Эта процедура не предполагает у рекурсивынх и полномочных серверов DNS соответствия данной спецификации или осведомлённости о SVCB.

Клиент считается независимым от SVCB (SVCB-optional), если он может соединяться без применения записей ServiceMode, иначе его называбт зависимым от SVCB (SVCB-reliant). Клиентам для уже существующих протоколов (например, HTTP) **нужно** реализовать независимое от SVCB поведение (за исключением случаев, указанных в параграфе 3.1 и при изменении спецификаций в будущем).

Независимым от SVCB клиентам **следует** параллельно вводить любые другие запросы DNS, которые могут потребоваться при отсутствии записи SVCB, чтобы минимизировать задержку и обеспечить оптимизацию, описанную в разделе 5. По завершении распознавания SVCB (независимо от результата) с обработкой хотя бы одной записи AliasMode таким клиентам **нужно** добавить в конец списка конечных точек точку, состоящую из финального значения $QNAME и номера порта полномочной конечной точки, но без SvcParams. К этой конечной точке будут обращаться перед откатом к режимам соединения без SVCB, что позволит клиентам использовать запись AliasMode, где TargetName имеет записи A и/или AAAA, но не имеет записей SVCB. Клиент организует соединение с использованием списка конечных точек. Ему **следует** начинать с высокоприоритетных точек, используя менее приоритетные при неудаче. Клиент распознаёт записи AAAA и/или A для выбранного TargetName и **может** выбирать между ними, используя такой подход, как Happy Eyeballs [HappyEyeballsV2]. При завершении списка без удачного соединения клиент пытается использовать режимы соединения без SVCB.

В разделе 5 рассмотрена оптимизация, позволяющая избежать дополнительной задержки по сравнению с обычным поиском AAAA/A.

3.1. Обработка отказов при распознавании

Если отклики DNS криптографически защищены (например, DNSSEC или TLS [DoT] [DoH]) и распознавание SVCB завершается отказом по ошибке аутентификации, отклику SERVFAIL, транспортной ошибке или тайм-ауту, клиенту **следует** прервать попытку доступа к сервису, даже если он независим от SVCB. Иначе активный злоумышленник сможет организовать атаку на понижение, препятствуя доступу пользователя к SvcParams.

Ошибка SERVFAIL может возникать, если домен использует подпись DNSSEC, рекурсивный сервер проверяет DNSSEC, а злоумышленник находится между рекурсивным распознавателем и полномочным сервером DNS. Транспортные ошибки и тайм-ауты могут возникать при размещении атакующего между клиентом и рекурсивным распознавателем с возможностью селективного отбрасывания запросов или откликов SVCB по их размеру или иным наблюдаемым параметрам.

Если клиент проверяет DNSSEC откликов A/AAAA, ему **следует** применять такую же проверку для SVCB. Иначе атакующий сможет обойти защиту A/AAAA, подделывая отклики SVCB, направляющие клиента на другой IP-адрес.

Если отклики DNS не защищены криптографически, клиент **может** считать отказ при распознавании SVCB критичным или некритичным.

Если клиент не может завершить распознавание SVCB из-за ограничения длины цепочки, он **должен** вернуться к полномочной конечной точке, как при отсутствии записи SVCB.

3.2. Клиенты, использующие прокси

Клиенты, использующие ориентированные на домен транспортные прокси, такие как HTTP CONNECT ([RFC7231], параграф 4.3.6) и SOCKS5 [RFC1928], имеют возможность работать с именованными адресатами и им не нужно делать запросов A или AAAA для целевых доменов. Если клиент настроен на работу с именованными адресатами, а прокси не поддерживает запросы SVCB (например, через связанный распознаватель DNS), клиенту придётся отдельно выполнять распознавание SVCB, что может приводить к раскрытию адресатов дополнительным сторонам, а не только прокси. Клиентам в такой конфигурации **следует** организовать отдельную процедуру распознавания SVCB с подобающей защитой приватности. Если это невозможно, независимые от SVCB клиенты **должны** полностью отключить распознавание SVCB, а клиенты с зависимостью от SVCB **должны** считать конфигурацию непригодной.

Если клиент не использует SVCB и именованных адресатов, ему **следует** соблюдать стандартный процесс распознавания SVCB, выбирая вариант с наименьшим SvcPriority, совместимый с клиентом и прокси. При соединении с использованием записи SVCB клиент **должен** предоставить прокси финальное значение TargetName и порт для выполнения требуемого поиска A и AAAA. Такой подход обеспечивает несколько преимуществ, указанных ниже.

- По сравнению с отключением SVCB:
  - клиент может использовать SvcParams (при наличии), которые пригодны лишь для конкретной цели TargetName и могут включать сведения, повышающие производительность (например, поддерживаемые протокоы) и улучшающие приватность;
  - сервис на вершине домена может использовать псевдонимы.
- По сравнению с предоставлением прокси IP-адреса:
  - прокси может выбирать между адресами IPv4 и IPv6 для сервера в соответствии с его конфигурацией;
  - гарантирует получение прокси адресов на основе географического положения, а не по клиентам;
  - обеспечивается более быстрый откат для адресатов TCP с несколькими адресами из одного семейства.

4. Поведение серверов DNS

4.1. Полномочный сервер

При ответе на запрос SVCB полномочному серверу DNS **следует** возвращать записи A, AAAA и SVCB в разделе Additional для любых TargetName в зоне. Если зона подписана, серверу **следует** включать в раздел Additional записи DNSSEC, подтверждающие наличие или отсутствие указанных записей. Исключения указаны в параграфе 4.4.

4.2. Рекурсивный распознаватель

Независимо от осведомлённости распознавателя о SVCB, обычный процесс, используемый для неизвестных типов RR [RFC3597], создаёт раздел Answer в отклике. Рекурсивным распознавателям, понимающим SVCB, **следует** помогать клиенту выполнить процедуру раздела 3 с минимальной общей задержкой, включив дополнительную полезную информацию в раздел Additional, как указано ниже.

1. Включить результаты распознавания SVCB. Если достигнут локальный предел размера цепочки у распознавателя (может отличаться от предельного размера у клиента), процесс завершается.
2. Если какие-либо из распознанных записей SVCB имеют режим AliasMode, среди них выполняется случайный выбор и для TargetName выбранной записи распознаются SVCB, A и AAAA.
   - Если распознана какая-либо из записей SVCB, возврат к п. 1.
   - Иначе результаты распознавания A и AAAA включаются в отклик и процесс завершается.
3. Если все распознанные записи SVCB имеют режим ServiceMode, распознаются A и AAAA для каждого TargetName (или для имени владельца, если TargetName = .), включаются в отклик и процесс завершается.

В этой процедуре распознавание означает обычную рекурсивную процедуру как при обработке запроса для RRset и включает следование всем псевдонимам, которым обычно следует распознаватель (например, CNAME, DNAME [DNAME]). Ошибки и аномалии при получении дополнительных записей **могут** прерывать процесс, но им **недопустимо**

самим по себе заставлять распознаватель передавать отклик с отказом. В параграфе 2.4.2 указаны дополнительные меры защиты рекурсивных распознавателей от петель, а в параграфе 5.2 - возможные меры оптимизации процедуры.

4.2.1. DNS64

Распознаватели DNS64 синтезируют отклики на запросы AAAA для имён, имеющих только запись A (параграф 5.1.7 в [RFC6147]). Понимающим SVCB распознавателям DNS64 **следует** применять ту же логику синтеза при распознавании записей AAAA для TargetName, включаемых в раздел Additional (п. 2 в параграфе 4.2) и **можно** опускать в разделе записи A.

Распознавателям DNS64 **недопустимо** переносить логику синтеза AAAA на подсказки IP в SvcParams (параграф 7.3). Изменение подсказок IP помешает проверке DNSSEC для записи SVCB и не повысит производительность выполнения приведённой выше рекомендации.

4.3. Общие требования

Рекурсивные распознаватели должны быть способны передавать записи SVCB с нераспознанными SvcParamKey. Это **можно** сделать, считая всю часть SvcParams в записи необрабатываемой (opaque), даже если её содержимое недействительно. Если за понятным SvcParamKey следует значение, которое недействительно в соответствии со спецификацией SvcParam, рекурсивный распознаватель **может** сообщить об ошибке (такой как SERVFAIL) вместо возврата записи. Для комплексных типов, интерпретация которых может зависеть от реализации или в будущем могут быть добавлены разрешённые значения (например, URI или alpn), распознавателям **следует** ограничивать проверки заданными пределами.

При отклике на запрос с битом DNSSEC OK [RFC3225] поддерживающие DNSSEC рекурсивные распознаватели и полномочные серверы DNS **должны** сопровождать каждый набор RRset в разделе Additional теми же связанными с DNSSEC записями, которые были бы переданы с этим RRset в разделе Answer (например, RRSIG, NSEC, NSEC3).

В соответствии с параграфом 5.4.1 в [RFC2181]: «Непроверенные RR полученные и кэшированные из ... дополнительного раздела ... не следует кэшировать таким образом, чтобы они могли возвращаться как ответы на полученные запросы. Они могут возвращаться как дополнительная информация, когда это приемлемо». Следовательно, рекурсивные распознаватели **могут** кэшировать записи из раздела Additional для использования при заполнении раздела Additional в откликах и для общего пользования, если они аутентифицированы DNSSEC.

4.4.Подсеть клиента EDNS

Опция подсети клиента EDNS (EDNS Client Subnet или ECS) [RFC7871] позволяет рекурсивным распознавателям запрашивать адреса IP, подходящие для определённого диапазона адресов клиента. Записи SVCB могут содержать адреса IP (в SvcParams ipv*hint) или направлять пользователй к зависящему от подсети TargetName, поэтому рекурсивным распознавателям **следует** включать в запросы SVCB ту же опцию ECS, что и в запросы A/AAAA.

В параграфе 7.3.1 [RFC7871] сказано: «Любые записи из [раздела Additional] **недопустимо** привязывать к сети». Поэтому при обработке запроса с QTYPE, совместимым с SVCB, распознавателям **следует** считать любые записи из раздела Additional, имеющими SOURCE PREFIX-LENGTH = 0 и SCOPE PREFIX-LENGTH, как указано в опции ECS. Полномочные серверы **должны** опускать такие записи, если они не подходят для использования любыми оконечными распознавателями (stub), установившими SOURCE PREFIX-LENGTH = 0. Это заставит распознаватель выполнить следующий (follow-up) запрос, который может вернуть корректно установленную опцию ECS (это похоже на использование CNAME с опцией ECS, описанное в параграфе 7.2.1 [RFC7871]). Полномочные серверы, опускающие записи Additional, могут избежать дополнительной задержки из-за повторного запроса, следуя рекомендации параграфа 10.2.

5. Оптимизация производительности

Для оптимальной производительности (минимального времени организации соединения) клиентам **следует** реализовать кэш DNS на своей стороне. Отклики в разделе Additional отклика SVCB **следует** помещать в кэш до выполнения последудующих запросов. Такое поведение и соответствующие спецификации серверы DNS предотвратят дополнительные задержки при организации соединения с использованием SVCB.

Для повышения производительности при использовании не соответствующих спецификации распознавателей клиенту **следует** вводить запросы A и/или AAAA параллельно каждому запросу SVCB на основе предсказания значения TargetName (см. параграф 10.2).

После получения ServiceMode RRset клиент **может** попробавать несколько опций параллельно, а также **может** заранее получить записи A и AAAA для нескольких TargetName.

5.1. Оптимистическое соединение заранее и повторное использование

Если отклик с адресом приходит до соответствующего отклика SVCB, клиент **может** инициировать соединение, как будто запрос SVCB вернул NODATA, но **недопустимо** передавать какую-либо информацию, которая может быть изменена в результате получения отклика SVCB. Например, в будущем могут быть определены SvcParamKey, меняющие TLS ClientHello.

Клиентам, реализующим такую оптимизацию, **следует** выждать 50 мсек до начала оптимистического соеднения заранее, как рекомендовано в [HappyEyeballsV2].

Запись SVCB согласуется с соединением, если клиент будет пытаться организовать с её помощью эквивалентное соединение. Если запись SVCB согласуется с активным или создаваемым соединением C, клиент **может** выбрать эту запись и использовать C как своё соединение. Предположим, что клиент получил для TLS через TCP SVCB RRset

```
_1234._bar.example.com. 300 IN SVCB 1 svc1.example.net. (
    ipv6hint=2001:db8::1 port=1234 )
                           SVCB 2 svc2.example.net. (
    ipv6hint=2001:db8::2 port=1234 )
```

Если клиент находится в состоянии организации соединения TCP с [2001:db8::2]:1234, он **может** продолжить с TLS на этом соединении даже при наличии в RRset записи с более высоким приоритетом. Если ни одна из записей SVCB не согласуется с активным или организуемым соединением, клиент переходит к организации соединения в соответствии с разделом 3.

5.2. Генерация и использование неполных откликов

При выполнении процедуры из параграфа 4.2 рекурсивные распознаватели **могут** прерывать её заранее и создавать отклик, где опущены некоторые из связанных RRset. Это **требуется** при достижении предела длины цепочки (п. 1 в параграфе 4.2), но может быть уместно при достижении максимального размера отклика или при повышении производительности за счёт ответа до полного соблюдения зависимостей. При исключении некоторых RRset рекурсивным распознавателям **следует** отдавать приритет записям с меньшим SvcPriority.

Как указано в разделе 3, клиенты **должны** быть способны получить дополнительную информацию, требуемую для применения записи SVCB, если сведения не включены в исходный отклик. Для оптимизации производительности клиент **может** предпочесть записи SVCB, не требующие дополнительных запросов DNS, независимо от их приоритета.

6. Совместимые с SVCB типы RR

Тип RR считается совместимым с SVCB, если он допускает реализацию, идентичную SVCB в части:

- формата представления RDATA;
- формата передачи RDATA;
- рестра IANA для SvcParamKey;
- обработки полномочным сервером раздела Additional;
- процесса рекурсивного распознавания;
- класса, к которому тип относится (Internet, IN [RFC1035]).

Это позволяет полномочным и рекурсивным серверам DNS применять идентичную обработку совместимых с SVCB типов RR.

Остальное поведение, описанное как применимое к SVCB RR, применимо и к совместимым с SVCB типам RR, если явно не указано иное. При следовании записи AliasMode (параграф 2.4.2) типа $T последующий запрос TargetName также **должен** выполняться для типа $T.

Этот документ определяет 1 совместимый с SVCB тип RR (кроме самой SVCB) - HTTPS RR (раздел 9), который позволяет избежать префиксных меток Attrleaf [Attrleaf] для улучшения совместимости с шаблонами и CNAME, широко применяемыми в HTTP.

Авторам стандартов следует тщательно продумать, применять ли SVCB или определить новый тип, совместимый с SVCB, поскольку выбор трудно будет изменить после внедрения.

7. Исходные SvcParamKey

Здесь определено несколько исходных SvcParamKey. Эти ключи полезны для схемы https и ожидается, что большинство из них будет применимо и с другими схемами.

Каждый документ для нового отображения протокола **должен** задавать, какие ключи применимы и безопасны для использования. Отображения протоколов **могут** изменять интерпретацию SvcParamKey, но **недопустимо** изменение форматов представления и передачи.

7.1. alpn и no-default-alpn

SvcParamKey alpn и no-default-alpn совместно указывают набор идентификаторов согласования протокола прикладного уровня (Application-Layer Protocol Negotiation или ALPN) [ALPN] и связанных транспортных протоколов, поддерживаемых данной конечной точкой сервиса (набор SVCB ALPN).

Как и в Alt-Svc [AltSvc], каждый идентификатор протокола ALPN служит для указания прикладного протокола и связанного набора протоколов, поддерживаемых конечной точкой (стек протоколов). Присутствие идентификатора протокола ALPN в наборе SVCB ALPN указывает, что конечная точка сервиса, описываемая TargetName и другими параметрами (например, port), предоставляет услуги со стеком протоколов, связанным с этим идентификатором ALPN.

Клиенты фильтруют набор идентификаторов ALPN в соответствии с набором поддерживаемых клиентом протоколов и это сообщает о применяемом базовом транспортном протоколе (таком как QUIC через UDP или TLS через TCP). Идентификаторы протоколов ALPN, которые не указывают однозначно стек протоколов (например Identification Sequence может применяться как для TLS, так и для DTLS), не совместимы с этим SvcParamKey и их **недопустимо** включать в набор SVCB ALPN.

7.1.1. Представление

ALPN указываются зарегистрированной идентификационной последовательностью (Identification Sequence, alpn-id) размером от 1 до 255 октетов.

```
alpn-id = 1*255OCTET
```

Для alpn значению представления **нужно** быть списком (через запятую, Приложение A.1) из одного или нескольких alpn-id. Реализации файлов зон **могут** запрещать символы , и \ в ALPN ID вместо реализации процедуры экранирования списка значений с помощью специального ключа (например, key1=\002h2), если такие символы нужны.

В формате передачи alpn состоит хотя бы из одного alpn-id с префиксом размера (1 октет) и эти парф размер-значение объединяются (конкатенация) в SvcParamValue. Пары **должны** точно заполнить SvcParamValue, иначе значение будет некорректно сформировано.

Значение no-default-alpn в формате представления и передачи **должно** быть пустым. При наличии no-default-alpn в RR запись должна включать и alpn для самосогласованности (параграф 2.4.3).

Каждая схема, использующая SvcParamKey, задаёт принятый по умолчанию набор ALPN ID, который поддерживается почти всеми клиентами. Этот набор **может** быть пустым. Определение набора SVCB ALPN клиент начинает со списка alpn-id из SvcParamKey alpn и добавляет принятый по умолчанию набор, если нет SvcParamKey no-default-alpn.

7.1.2. Использование

Для организации соединения с конечной точкой клиент **должен** выполнить указанные ниже действия.

1. Пусть SVCB-ALPN-Intersection - набор протоколов в наборе SVCB ALPN, поддерживаемых клиентом.
2. Пусть Intersection-Transports - набор вариантов транспорта (например, TLS, DTLS, QUIC), подразумеваемых протоколами SVCB-ALPN-Intersection.
3. Для каждого варианта транспорта из Intersection-Transports создаётся список ProtocolNameList, содержащий Identification Sequence для всех поддерживаемых клиентом для этого транспорта протоколов ALPN, независимо от набора SVCB ALPN.

Например, если набор SVCB ALPN - это [http/1.1, h3], а клиент поддерживает HTTP/1.1, HTTP/2 и HTTP/3, он может попытаться соединиться с использованием TLS через TCP с ProtocolNameList [http/1.1, h2], а также с использованием QUIC с ProtocolNameList [h3].

После создания клиентом ClientHello происходит согласование протокола, как указано в [ALPN], без учёта набора SVCB ALPN.

Клиент **может** реализовать процедуру отката, используя менее предпочтительный транспорт, если не удалось соединиться по более предпочтительному. Поддержка отката уязвима к атакам со стороны злоумышленников в сети, блокирующих более предпочтительный транспорт, но может потребоваться для совместимости с имеющимися сетями.

При этой процедуре злоумышленник, способный менять DNS и сетевой трафик, может воспрепятствовать успешному транспортному соединению, но не сможет как-то иначе повлиять на выбор протокола ALPN. Процедура также гарантирует, что в каждом ProtocolNameList будет хотя бы 1 протокол из набора SVCB ALPN.

Клиенту **не следует** пытаться соединиться с конечной точкой сервиса, чей набор SVCB ALPN не содержит поддерживаемых протоколов.

Для обеспечения согласованного поведения клиент **может** отвергнуть SVCB RRset целиком и вернуться к базовой организации соединения, когда все совместимые RR указывают no-default-alpn, даже если соединение может быть успешным с использованием протокола ALPN, не выбранного по умолчанию.

Оператору зоны **следует** убедиться, что хотя бы одна RR в каждом RRset поддерживает принятый по умолчанию транспорт. Это обеспечивает совместимость с наибольшим числом клиентов.

7.2. port

Ключ SvcParamKey port указывает порт TCP или UDP, который следует использовать для доступа к дополнительной конечной точке. При отсутствии этого ключа клиенту **нужно** использовать порт полномочной конечной точки.

SvcParamValue представляется одним десяьтчным числом от 0 до 65535 в кодировке ASCII. Любое другое значение (например, пустое) является синтаксической ошибкой. Для упрощения разбора в SvcParamValue **недопустимо** включать экранирующие последовательности.

В формате передачи SvcParamValue указывается 2-октетным целым числом с сетевым порядком байтов.

Если между клиентом и конечной точкой сервиса имеется межсетевой экран, ограничивающий набор портов, смена номера порта может воспрепятствовать доступу клиента к службе, поэтому операторам следует соблюдать осторожность при использовании этого SvcParamKey для задания портов, отличных от принятого по умолчанию.

7.3. ipv4hint и ipv6hint

Ключи ipv4hint и ipv6hint передают адреса IP, которые клиент **может** использовать для доступа к сервису. Если записи A и AAAA для TargetName доступны локально, клиенту **следует** игнорировать подсказки адресов, в ином случае ему **следует** выполнить запросы A и/или AAAA для TargetName в соответствии с разделом 3 и адреса IP из этих откликов клиенту **следует** применять для будущих соединений. Клиенты **могут** прерывать любые соединения, использующие адреса из подсказок, и переключаться на адреса из отклика на запрос TargetName. Отказ от использования адресов из откликов A и/или AAAA может негативно влиять на распределение нагрузки и другие функции, связанные с местоположением, снижая производительность работы клиента.

В формате представления значение **нужно** указывать списком через запятые (Приложение A.1) из одного или нескольких IP-адресов подходящего семейства в стандартном текстовом формате [RFC5952] [RFC4001]. Для упрощения разбора в SvcParamValue **недопустимо** включать экранирующие последовательности.

В формате передачи каждый параметр является последовательностью адресов IP с сетевым порядком байтов (в соответствии с семейством адресов). Как и в RRset A и AAAA списокк является неупорядоченным и клиентам **следует** выбирать адрес из списка в случайном порядке. Пустой список адресов является недействительным.

При выборе между адресами IPv4 и IPv6 клиенты могут применять подход, аналогичный Happy Eyeballs [HappyEyeballsV2]. При наличии лишь ipv4hint клиенты NAT64 могут синтезировать адреса IPv6, как описано в [RFC7050], или игнорировать ключ ipv4hint и ждать распознавания AAAA (раздел 3). Для лучшей производительности операторам серверов **следует** включать параметр ipv6hint, если имеется ipv4hint.

Эти параметры предназначены для минимизации дополнительной задержки при организации соединения, когда рекурсивный распознаватель не соответствует требованиям раздела 4, и их **не следует** включать, если большинство клиентов пользуются соответствующими требованиям распознавателями. Если TargetName - это имя службы или

владельца (которое можно указать как .), оператору сервера **недопустимо** включать эти подсказки, поскольку они вряд ли дадут выигрыш в производительности.

7.4. mandatory
См. раздел 8.

8. Совместимость ServiceMode RR и обязательные ключи
В ServiceMode RR ключ SvcParamKey считается обязательным (mandatory), если RR не будет корректно работать у клиентов, игнорирующих этот SvcParamKey. В каждом отображении протоколв SVCB **следует** задавать набор ключей, которые «автоматически обязательны», т. е. являются обязательными при наличии в RR. SvcParamKey mandatory применяется для указания любых обязательных ключей данной RR в дополнение к имеющимся автоматически обязательным.

ServiceMode RR считается совместимой (compatible) с клиентом, если тот понимает все обязательные ключи и их значения указывают возможность успешной организации соединения. Несовместиые RR игнорируются (см. п. 5 в процедуре из раздела 3).

Значениям в формате представления **нужно** быть списком через запятые (Приложение A.1) из одного или нескольких SvcParamKey с их зарегистрированными именами или в формате неизвестного ключа (unknown-key, параграф 2.1). Ключи **можно** указывать в любом порядке, но **недопустимо** повторение какого-либо ключа. Для самосогласованности (параграф 2.4.3) перечисленные ключи **должны** также присутствовать в SvcParams.

Для упрощения синтаксического анализ в данное значение SvcParamValue **недопустимо** включать экранирующие последовательности.

Ниже приведён пример действительного списка SvcParams.

```
ipv6hint=... key65333=ex1 key65444=ex2 mandatory=key65444,ipv6hint
```
В формате передачи ключи указываются численными значениями с сетевым порядком байтов, объединёнными (конкатенация) строго с порядке роста номеров.

Данный ключ SvcParamKey всегда автоматически обязателен и **недопустимо** его присутствие в своём списке значений. Другие автоматически обязательные ключи **не следует** включать в список (это увеличивает размер, не давая никакого эффекта).

9. Использование привязки сервиса к HTTP
Для использования любого протокола с SVCB требуется спецификация соответствующего отображения. В этом разделе задано отображение для схем URI http и https [HTTP].

Для обеспечения специальной обработки при использовании HTTP определён тип HTTPS RR как совместимый с SVCB тип RR, связанный со схемами https и http. Клиентам **недопустимо** выполнять запросы и принимать отклики SVCB для схем https и http.

В формате представления запись имеет вид

```
Name TTL IN HTTPS SvcPriority TargetName SvcParams
```
Все SvcParamKey, заданные в разделе 7, можно применять в HTTPS RR. Принятый по умолчанию набор ALPN ID включает одно значение http/1.1. Автоматически обязательными ключами (раздел 8) являются port и no-default-alpn (как указано в разделе 8, клиенты должны реализовать эти ключи или отвергать любые RR, где они есть). Клиентам, ограничивающим порты получателя в https URI (например, с помощью списка «плохих портов» из [FETCH]), **следует** применять те же ограничения к SvcParamKey port.

Наличие HTTPS RR для источника (origin) также указывает, что клиенту следует подключаться с защитой и применять схему https, как указано в параграфе 9.5. Это позволяет применять HTTPS RR к уже имеющимся URL со схемой http, гарантируя, что клиент использует защищённое и аутентифицированное соединение.

HTTPS RR соответствует концепциям, введенным в HTTP Alternative Services [AltSvc]. Клиентам и серверам, реализующим HTTPS RR, не требуется реализовать Alt-Svc.

9.1. Имена в запросах HTTPS RR
В HTTPS RR применяется префиксное именование портов (параграф 2.3) с одним изменением - если используется схема https и порт 443, исходное значение QNAME от клиента совпадает с именем службы (т. е. именем хоста-источника) без префиксной метки. Благодаря удалению меток Attrleaf [Attrleaf], используемых в SVCB, эта конструкция поддерживает автономное (offline) подписывание DNSSEC доменов с шаблонами, которые обычно применяются в HTTP. Использование имени службы в качестве имени владельца записи HTTPS без префикса также позволяет целям имеющихся цепочек CNAME (например, хостам CDN) начинать возврат откликов HTTPS RR, не требуя от домена источника настройки и поддержки дополнительного делегирования.

Процесс следования HTTPS RR AliasMode и псевдонимам CNAME не отличается от SVCB (параграф 2.4.2 и раздел 3).

Клиенты всегда преобразуют URL http в https до выполнения запроса HTTPS RR с помощью процесса из параграфа 9.5, поэтому владельцам доменов **недопустимо** публикова HTTPS RR с префиксом _http.

Отметим, что ни одна из форм не меняет источник (origin) или полномочия (authority) HTTPS. Например, клиенты **должны** продолжать проверку сертификатов TLS для хостов на основе origin.

9.2. Сравнение с Alt-Svc
Публикация HTTPS RR ServiceMode в DNS должна быть похода на передачу поля Alt-Svc через HTTP, а получение HTTPS RR - на получение Alt-Svc через HTTP. Однако имеются различия в предполагаемом поведении клиентов и серверов.

9.2.1. Использование ALPN

В отличие от значений поля Alt-Svc, HTTPS RR могут включать несколько ALPN ID. Назначение и использование идентификаторов описано в параграфе 7.1.2.

9.2.2. Недоверенные каналы

Записи HTTPS не требуют и не предоставляют проверки подлинности (подписи DNSSEC и их проверка для аутентификации являются **необязательными**). Процесс распознавания DNS моделируется как недоверенный канал, который может контролироваться злоумышленником, поэтому параметрам Alt-Svc, которые в этой модели не могут быть защищены, **недопустимо** иметь соответствующие SvcParamKey. Например, нет SvcParamKey, соответствующего параметру Alt-Svc persist, поскольку этот параметр небезопасно передавать через недоверенный канал.

9.2.3. Срок действия кэша

Не существует SvcParamKey, соответствующего параметру Alt-Svc ma (максимальный возраст) и вместо этого операторы серверов представляют завершение срока действия в DNS TTL. Подходящее значение TTL может отличаться значения ma, используемого для Alt-Svc, в зависимости от желаемой эффективности и гибкости. Некоторые кэши DNS некорректно увеличивают срок действия записей DNS сверх указанного TTL, поэтому операторы серверов не могут полагаться на своевременное завершение срока действия HTTPS RR. Сокращать TTL для компенсации некорректного кэширования **не рекомендуется**, поскольку это снижает эффективность корректно работающих кэшей и не гарантирует быстрого завершения срока действия из некорректных кэшей. Вместо этого операторам серверов **следует** поддерживать совместимость с просроченными записями, пока не будет ясно, что почти все соединения перешли на новую конфигурацию.

9.2.4. Детализация

Отправка Alt-Svc через HTTP позволяет серверу приспособить значение поля Alt-Svc специально для клиента. При использовании HTTPS RR группы клиентов обязательно получают одинаковые SvcParams, поэтому HTTPS RR не подходят для случаев, требующих детализации для каждого клиента.

9.3. Взаимодействие с Alt-Svc

Клиентам, поддерживающим записи Alt-Svc и HTTPS и организующим соединения на основе кэшированного отклика Alt-Svc, **следует** извлекать любые записи HTTPS для Alt-Svc alt-authority и убеждаться в соответствии их попыток соединения как параметрам Alt-Svc, так и всем полученным HTTPS SvcParams. При наличии в записи HTTPS порта и TargetName они применяются для организации соединения (в соответствии с разделом 3). Предположим, например, что https://example.com передаёт в поле Alt-Svc значение

```
Alt-Svc: h2="alt.example:443", h2="alt2.example:443", h3=":8443"
```

Клиент будет извлекать записи HTTPS

```
alt.example.              IN HTTPS 1 . alpn=h2,h3 foo=...
alt2.example.             IN HTTPS 1 alt2b.example. alpn=h3 foo=...
_8443._https.example.com. IN HTTPS 1 alt3.example. (
    port=9443 alpn=h2,h3 foo=... )
```

На основе этих сведений всегда будут разрешены попытки указанных ниже соединений.

- HTTP/2 с alt.example:443.
- HTTP/3 с alt3.example:9443.
- Откат клиента к соединению без Alt-Svc.

Указанные ниже попытки соединений не будут разрешены.

- HTTP/3 с alt.example:443 (не соответствует Alt-Svc).
- Любые соединения с alt2b.example (нет ALPN ID, согласующегося с записью HTTPS и Alt-Svc).
- HTTPS через любой порт TCP с alt3.example (не соответствует Alt-Svc).

Предположим, что foo - это ключ SvcParamKey, делающим клиента основанным на SVCB. Указанные ниже попытки соединений на основе лишь Alt-Svc будут разрешены только в случае отсутствия у клиента поддержки foo, поскольку они основаны на необязательном в SVCB откате.

- HTTP/2 с alt2.example:443.
- HTTP/3 с example.com:8443.

В alt-authority **следует** передавать те же SvcParams, что и origin, если нет уверенности в безопасности отклонения. Как отмечено в параграфе 2.4 [AltSvc], клиенты **могут** запрещать любое соединение Alt-Svc на основе своих критериев, например, при отсутствии функций защиты приватности, доступных на полномочной (authority) конечной точке.

9.4. Требование указания имени сервера

Клиенту **недопустимо** применять отклик HTTPS RR, если он не поддерживает расширение TLS для указания имени сервера (Server Name Indication или SNI) и не указывает имя источника в TLS ClientHello (оно может шифроваться в соответствии с будущими спецификациями, например, [ECH]). Это поддерживает сохранение адресов IP.

Отметим, что TLS SNI (а также HTTP Host или :authority) указывают источник, а не TargetName.

9.5. Строгая транспортная защита HTTP (HSTS)

An HTTPS RR направляет клиента на взаимодействие с данным хостом только через защищённый транспорт, подобно HSTS [HSTS]. Перед запросом по схеме http клиенту **следует** выполнить поиск для определения наличия записи HTTPS RR для этого источника. Для этого клиенту **следует** создать URL https, как указано ниже.

1. Схема http заменяется на https.
2. Если URL http явно задаёт порт 80, указывается порт 443.
3. Остальные аспекты URL не изменяются.

Эта конструкция эквивалентна п. 5 из параграфа 8.3 в [HSTS].

Если запрос HTTPS RR для этого URL https возвращает какие-либо HTTPS RR AliasMode или совместимые HTTPS RR ServiceMode (см. раздел 8), клиенту **следует** вести себя как при получении кода статуса HTTP 307 (Temporary Redirect - временное перенаправление) с этим URL https в поле Location (получение несовместимой записи ServiceMode не ведёт к перенаправлению). Поскольку HTTPS RR часто передаются по незащищённому каналу (DNS), клиенту **недопустимо** доверять этому сигналу больше, чем при получении отклика 307 (Temporary Redirect) через HTTP без шифрования.

Публикация HTTPS RR может приводить к неожиданным результатам или потере функциональности в случаях, когда ресурс http не перенаправляет на https и не указывает на тот же базовый ресурс.

При отказе в соединении https из-за ошибки базового защищённого транспорта, такой как отказ при проверке сертификатов, некоторые клиенты в настоящее время предлагают «пользовательский выход» (user recourse), которое позволяет обойти ошибку защиты и подключиться. При выполнении запроса по схеме https к источнику с HTTPS RR напрямую или через описанное выше перенаправления, клиент **может** исключить вариант «пользовательского выхода». Поэтому источникам, публикующим HTTPS RR, **недопустимо** полагаться на «пользовательский выход» для доступа. Дополнительные сведения приведены в параграфах 8.4 и 12.1 [HSTS].

9.6. Использование HTTPS RR в других протоколах

Все соединения HTTP с именованными источниками могут использовать HTTPS RR, даже когда HTTP является частью другого протокола или используется без явной схемы URI, основанной на HTTP (параграф 4.2 в [HTTP]). Например, клиентам, поддерживающим HTTPS RR и реализующим [WebSocket] с использованием изменённого согласования при открытии из [FETCH-WEBSOCKETS], **следует** применять HTTPS RR для requestURL.

При использовании HTTP в контексте, где URL и перенаправление неприменимы (например, соединения с HTTP-прокси), клиентам, нашедшим соответствующую HTTPS RR, **следует** реализовать повышение уровня защиты, эквивалентное указанному в параграфе 9.5.

Такие протоколы **могут** определять свои отображения SVCB, которые **могут** иметь преимущество перед HTTPS RR.

10. Структура зоны

10.1. Структурирование зон для гибкости

Каждый набор RRset ServiceMode может обслуживать лишь одну схему, указываемую именем владельца и типом RR. Для базового типа SVCB RR это означает, что каждое имя владельца может быть использовано лишь в одной схеме. Требование префикса с подчёркиванием (параграф 2.3) гарантирует соблюдение этого для исходного запроса, но владельцы зон обязаны выбирать имена, соответствующие этому требованию, при использовании псевдонимов, включая записи CNAME и AliasMode. При использовании базового типа SVCB RR с псевдонимами владельцам зон **следует** выбирать имя цели псевдонима, указывающее используемую схему (например, foosvc.example.net для схем foo). Это поможет избежать путаницы при необходимости добавления в конфигурацию другой схемы. При использовании нескольких номеров портов может оказаться полезным повторение префиксной метки в имени цели псевдонима (например, _1234._foo.svc.example.net).

10.2. Структурирование зон для производительности

Чтобы избежать задержки для клиентов, использующих не соответствующий спецификации распознаватель, владельцам доменов **следует** минимизировать применение записей AliasMode, а также **следует** выбирать TargetName в соответствии с предсказуемым соглашением, известным клиентам, чтобы те могли заранее отправлять запросы A и/или AAAA для TargetName (см. раздел 5). Если не задано иное, для начальной записи ServiceMode следует устанавливать в TargetName имя службы или ., если служба доступна через псевдоним.

```
$ORIGIN example.com. ; Источник
foo                  3600 IN CNAME foosvc.example.net.
_8080._foo.foo       3600 IN CNAME foosvc.example.net.
bar                   300 IN AAAA 2001:db8::2
_9090._bar.bar       3600 IN SVCB 1 bar key65444=...

$ORIGIN example.net. ; Зона сервис-провайдера
foosvc               3600 IN SVCB 1 . key65333=...
foosvc                300 IN AAAA 2001:db8::1
```

Рисунок 1. foo://foo.example.com:8080 доступен на foosvc.example.net, а bar://bar.example.com:9090 обслуживается локально.

Владельцам доменов **следует** избегать использования TargetName ниже DNAME, поскольку это, скорей всего, не требуется, а также замедляет и увеличивает отклик. **Не рекомендуется** также применять структуры зон, требующие следования более чем 8 псевдонимам (считая AliasMode и CNAME).

10.3. Вопросы эксплуатации

Некоторые полномочные серверы DNS могут не разрешать записи A или AAAA для имён, начинающихся с символа подчёркивания (см., например, [BIND-CHECK-NAMES]). Это может создавать проблемы при работе, если TargetName содержит метку Attrleaf, или при использовании в TargetName значения ., если имя владельца включает метку Attrleaf.

10.4. Примеры

10.4.1. Улучшения протокола

Рассмотрим простую зону, приведённую ниже.

```
$ORIGIN simple.example. ; Пример простой зоны
@ 300 IN A    192.0.2.1
        AAAA 2001:db8::1
```

Владелец домена может добавить запись

```
@ 7200 IN HTTPS 1 . alpn=h3
```

Эта запись будет указывать, что https://simple.example поддерживает QUIC в дополнение к HTTP/1.1 через TLS и TCP (принято по умолчанию). Запись может включать и другие сведения (например, нестандартный порт). Для https://simple.example:8443 запись имеет вид

```
_8443._https 7200 IN HTTPS 1 . alpn=h3
```

Эти записи указывают клиентам замену схемы на https при загрузке http://simple.example или http://simple.example:8443.

10.4.2. Псевдонимы на вершине зоны

Рассмотрим зону с псевдонимом CNAME

```
$ORIGIN aliased.example. ; Зона с услугами хостинга, где для
; субдоменов указаны псевдонимы высокопроизводительного пула серверов
www             7200 IN CNAME pool.svc.example.
; Вершина домена имеет фиксированные IP, поскольку CNAME не разрешены
@                300 IN A     192.0.2.1
                     IN AAAA  2001:db8::1
```

С помощью HTTPS RR владелец aliased.example может задать псевдоним на вершине, используя запись

```
@               7200 IN HTTPS 0 pool.svc.example.
```

При наличии этой записи понимающие HTTPS RR клиенты будут использовать один пул серверов для aliased.example и www.aliased.example (они также заменят http://aliased.example/... на https). Не поддерживающие HTTPS RR клиенты просто будут игнорировать новую запись.

Подобно CNAME, записи HTTPS RR не влияют на имя источника. При соединении клиенты продолжат воспринимать полномочные источники https://www.aliased.example и https://aliased.example, проверяя соответствующие сертификаты TLS.

10.4.3. Привязка параметров

Предположим, что основной пул серверов svc.example поддерживает HTTP/3, а резервный - не поддерживает. Это можно указать в виде

```
$ORIGIN svc.example. ; Провайдер хостинга
pool  7200 IN HTTPS 1 . alpn=h2,h3
              HTTPS 2 backup alpn=h2 port=8443
pool   300 IN A        192.0.2.2
              AAAA     2001:db8::2
backup 300 IN A        192.0.2.3
              AAAA     2001:db8::3
```

Эта конфигурация полностью совместима с примером «псевдонимов на вершине», независимо от поддержки клиентами HTTPS RR. Если клиент поддерживает HTTPS RR, все соединения перейдут на HTTPS, а клиенты будут использовать HTTP/3 (если могут). Параметры «привязаны» к каждому пулу серверов, поэтому каждый пул может использовать свой протокол, номер порта и т. п.

10.4.4. Конфигурация с несколькими CDN

Записи HTTPS RR предназначены для поддержки служб HTTPS, управляемых несколькими независимыми организациями, такими как провайдеры CDN или хостинга. Это включает случаи переноса сервиса от одного оператора к другому, а также мультиплексирования служб между несколькими операторами для повышения производительности, резервирования и т. п.

Приведённый ниже пример показывает конфигурацию, где www.customer.example возвращает разные отклики DNS для запросов в зависимости от времени или логики полномочного сервера DNS.

```
; Эта зона содержит и возвращает разные записи CNAME в разное время.
; RRset для www может включать лишь 1 запись CNAME.

; Иногда зона имеет
$ORIGIN customer.example.  ; Клиентский домен с несколькими CDN.
www 900 IN CNAME cdn1.svc1.example.

; в другое время
$ORIGIN customer.example.
www 900 IN CNAME customer.svc2.example.

; а в третье
$ORIGIN customer.example.
www 900 IN CNAME cdn3.svc3.example.

; Приведённые ниже сведения не меняются и включены всегда
$ORIGIN customer.example.  ; Клиентский домен с несколькими CDN.
; На вершине зоны задан псевдоним www для соответствия конфигурации.
@     7200 IN HTTPS 0 www
; Клиенты без поддержки HTTPS используют IP, не относящиеся к CDN.
```

```
                A    203.0.113.82
                AAAA 2001:db8:203::2

; Эти записи использует распознавание по пути cdn1.svc1.example.
; Эта сеть CDN использует свой дополнительный сервис для HTTP/3.
$ORIGIN svc1.example.  ; Домен для CDN 1.
cdn1     1800 IN HTTPS 1 h3pool alpn=h3
                 HTTPS 2 . alpn=h2
                 A    192.0.2.2
                 AAAA 2001:db8:192::4
h3pool 300 IN A 192.0.2.3
           AAAA 2001:db8:192:7::3

; Эти записи использует распознавание по пути customer.svc2.example.
; В этой сети CDN поддерживается только HTTP/2.
$ORIGIN svc2.example. ; Домен, где работает CDN 2.
customer 300 IN HTTPS 1 . alpn=h2
          60 IN A    198.51.100.2
                A    198.51.100.3
                A    198.51.100.4
                AAAA 2001:db8:198::7
                AAAA 2001:db8:198::12

; Эти записи использует распознавание по пути cdn3.svc3.example.
; В этой сети CDN нет записей HTTPS.
$ORIGIN svc3.example. ; Домен, где работает CDN 3.
cdn3      60 IN A    203.0.113.8
                AAAA 2001:db8:113::8
```

Отметим, что в приведённом выше примере конфигурации и возможности CDN различаются, но клиенты будут использовать HTTPS RR как единое целое.

Владельцам доменов следует быть осторожными в конфигурациях с несколькими CDN, поскольку с ними связан ряд описанных ниже сложностей.

- Если CDN 1 поддерживает желаемый протокол или свойство, а CDN 2 - нет, клиент будет уязвим к атакам на понижение со стороны злоумышленников в сети, которые заставят его получить записи CDN 2.
- Псевдонимы на субдомены на вершине зоны упрощают зону, но увеличивают время распознавания особенно при работе с не поддерживающими HTTPS рекурсивными распознавателями. Альтернативой может быть включение на вершине хоны псевдонима, напрямую указывающего имя, поддерживаемое CDN.
- Распознавание A, AAAA и HTTPS выполняется как независимые операции поиска, поэтому распознаватели могут наблюдать разные CNAME и следовать за ними в разные CDN. В результате клиенты могут обнаружить, что отклики A и AAAA не соответствуют TargetName в отклике HTTPS и им потребуются дополнительные запросы для получения корректных адресов IP. Включение ipv6hint и ipv4hint будет снижать влияние на производительность в таких случаях.
- Если не все CDN публикуют записи HTTPS, клиенты иногда будут получать для запросов HTTPS отклик NODATA (как для cdn3.svc3.example в примере), но получат записи A/AAAA из другой CDN. Клиент будет пытаться соединиться с этой CDN без использования преимуществ записей HTTPS.

10.4.5. Использование с другими протоколами

Для отличных от HTTP протоколов будут применяться запись SVCB RR и метка Attrleaf [Attrleaf]. Например, для доступа к ресурсу baz://api.example.com:8765 будет использоваться приведённая ниже запись SVCB, задающая псевдоним svc4-baz.example.net., который может, в свою очередь, возвращать записи AAAA/A и/или SVCB ServiceMode

```
_8765._baz.api.example.com. 7200 IN SVCB 0 svc4-baz.example.net.
```

HTTPS RR похожи на метки Attrleaf, если источник (origin) содержит отличный от принятого по умолчанию порт.

11. Взаимодействие с другими стандартами

Этот стандарт предназначен для сокращения задержки при соединениях и повышения уровня приватности пользователей. Операторам серверов, поддерживающих этот стандарт, **следует** реализовать также TLS 1.3 [RFC8446] и протокол сшивки статуса сертификатов (Online Certificate Status Protocol или OCSP) (т. е. запросы статуса сертификатов в соответствии с разделом 8 в [RFC6066]), которые обеспечивают существенный рост производительности и повышение приватности в сочетании с записями SVCB.

В плане реализации максимальных преимуществ в части приватности это предложение предназначено для использования на основе транспорта DNS с защитой приватности (такого как DNS over TLS [DoT] и DNS over HTTPS [DoH]). Однако повышение производительности и скромное улучшение приватности доступны и без этих стандартов.

Любая спецификация использования SVCB с протоколом **должна** иметь запись для этой схемы под типом SVCB RR в реестре IANA Underscored and Globally Scoped DNS Node Names [Attrleaf]. Схема **должна** иметь запись а реестре Uniform Resource Identifier (URI) Schemes [RFC7595] и спецификацию для применения с SVCB.

12. Вопросы безопасности

Записи SVCB/HTTPS RR можно распространять по недоверенным каналам, а от клиентов требуется проверять полномочность дополнительной конечной точки для сервиса (подобно параграфу 2.1 в [AltSvc]). Поэтому подписи и проверка DNSSEC являются необязательными для публикации и использования записей SVCB и HTTPS.

Клиенты должны убедиться, что их кэш DNS разделен по локальным сетям или очищается при смене сети для предотвращения внедрения локальным злоумышленником в одной из сетей поддельной записи DNS, позволяющей ему отслеживать пользователей или препятствовать их соединениям после выхода из данной сети.

Злоумышленник, способный воспрепятствовать распознаванию SVCB, может лишить клиентов связанных с этим преимуществ безопасности. Враждебный рекурсивный распознаватель может отказать в обслуживании запросов SVCB, а сетевые посредники часто тоже препятствуют распознаванию даже при проверке клиентами и рекурсивными распознавателями DNSSEC и использовании защищённого транспорта. Такие атаки с понижением могут препятствовать переходу на https, обеспечиваемому HTTPS RR (параграф 9.5) и отключить любые другие средства защиты, координируемые SvcParams. Для предотвращения этого параграф 3.1 рекомендует клиентам прерывать попытки организации соединения при обнаружении такой атаки.

Враждебный посредник DNS может подделать записи AliasMode "." (параграф 2.5.1) для блокировки доступа клиентов к определенным службам. Такой злоумышленник может блокировать целые домены, подделывая отклики с ошибками, но этот механизм позволяет ему нацеливаться на определённые протоколы или порты в домене. Клиентам, которые могут быть подвержены таким атакам, **следует** игнорировать записи AliasMode ".".

Враждебный посредник DNS или полномочный сервер может возвращать записи SVCB, указывающие любой IP-адрес и номер порта, включая адреса IP из локальной сети и номера портов, связанные с внутренними службами. Если атакующий может влиять на содержимое пакетов клиента (например, содержимое сеансовых квитанций TLS), а анализатор внутренней службы достаточно слаб, злоумышленник может получить доступ к внутренней службе (такие же проблемы возникают для записей SRV, а также перенаправления HTTP Alt-Svc и HTTP). В качестве смягчения последствий в документах по отображению SVCB **следует** указывать ограничения номеров портов, подходящие для поддерживаемого транспорта.

13. Вопросы приватности

Стандартные адресные запросы раскрывают намерения пользователя получить доступ к определённому домену. Эти сведения доступны рекурсивным распознавателям и многим другим сторонам при использовании открытого транспорта DNS. Запросы SVCB, как и запросы записей SRV и других типов RR, дополнительно раскрывают намерение пользователя применять определённый протокол. Обычно эта информация не является конфиденциальной, но её следует учитывать при добавлении поддержки SVCB в новом контексте.

14. Взаимодействие с IANA

14.1. Тип SVCB RR

Агентство IANA зарегистрировало новый тип DNS RR в реестре Resource Record (RR) TYPEs на странице Domain Name System (DNS) Parameters, как показано ниже.

```
Type:  SVCB
Value:  64
Meaning: Привязка службы общего назначения
Reference:  RFC 9460
```

14.2. Тип HTTPS RR

Агентство IANA зарегистрировало новый тип DNS RR в реестре Resource Record (RR) TYPEs на странице Domain Name System (DNS) Parameters, как показано ниже.

```
Type:  HTTPS
Value:  65
Meaning: Совместимый с SVCB тип для использования с HTTP
Reference:  RFC 9460
```

14.3. Новый реестр для параметров сервиса

Агентство IANA создало реестр Service Parameter Keys (SvcParamKeys) в категории Domain Name System (DNS) Parameters на новой странице DNS Service Bindings (SVCB). Этот реестр задаёт пространство имён для параметров, включая строковые представления и численные значения SvcParamKey. Этот реестр является совместным для совместимых с SVCB типов RR, таких как HTTPS RR.

14.3.1. Процедура

При регистрации **должны** указываться перечисленные ниже поля.

```
Number: Числовой идентификатор формата передачи (0-65535)
Name: Уникальное имя представления
Meaning: Краткое описание
Reference: Ссылка на спецификацию или источник регистрации
Change Controller: Человек или организация с контактными данными
```

В поле Name **должны** использоваться алфавитно-цифровые символы в нижнем регистре или - (параграф 2.1). **Недопустимы** имена, начинающиеся с key и invalid.

Новые записи регистрируются по процедуре Expert Review ([RFC8126], параграф 4.5). назначенный эксперт **должен** убедиться, что ссылка указывает стабильный и общедоступный документ, указывающий, как формат представления SvcParamValue преобразуется в формат передачи. Ссылка **может** указывать Internet-Draft или документ из любого другого источника с аналогичными гарантиями стабильности и доступности. Запись может иметь форму Same as (имя другого ключа) если в ней применяются такие же формат представления и передачи, как в существующей записи.

Такой порядок позволяет разрабатывать новые параметры, сохраняя функциональную совместимость файлов зон.

14.3.2. Исходное содержимое

Реестр Service Parameter Keys (SvcParamKeys) исходно содержит указанные в таблице 1 записи.

Таблица 1.

Номер	Имя	Назначение	Документ	Контролёр изменений
0	mandatory	Обязательные ключи данной RR	RFC 9460, раздел 8	IETF
1	alpn	Дополнительные поддерживаемые протоколы	RFC 9460, параграф 7.1	IETF
2	no-default-alpn	Нет поддержки принятого по умолчанию протокола	RFC 9460, параграф 7.1	IETF
3	port	Порт для дополнительной конечной точки	RFC 9460, параграф 7.2	IETF
4	ipv4hint	Подсказки адресов IPv4	RFC 9460, параграф 7.3	IETF
5	ech	Резерв (для Encrypted ClientHello)	-	IETF
6	ipv6hint	Подсказки адресов IPv6	RFC 9460, параграф 7.3	IETF
65280-65534	-	Резерв для приватного использования	RFC 9460	IETF
65535	-	Резерв (недействительный ключ)	RFC 9460	IETF

14.4. Другие обновления реестров

В соответствии с [Attrleaf] в реестр DNS Underscored and Globally Scoped DNS Node Names добавлена 1 запись.

Таблица 2.

Тип RR	Имя _NODE	Документ
HTTPS	_https	RFC 9460

15. Литература

15.1. Нормативные документы

[ALPN] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <https://www.rfc-editor.org/info/rfc7301>.

[Attrleaf] Crocker, D., "Scoped Interpretation of DNS Resource Records through "Underscored" Naming of Attribute Leaves", BCP 222, RFC 8552, DOI 10.17487/RFC8552, March 2019, <https://www.rfc-editor.org/info/rfc8552>.

[DoH] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

[DoT] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/info/rfc7858>.

[HappyEyeballsV2] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <https://www.rfc-editor.org/info/rfc8305>.

[HTTP] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC1928] Leech, M., Ganis, M., Lee, Y., Kuris, R., Koblas, D., and L. Jones, "SOCKS Protocol Version 5", RFC 1928, DOI 10.17487/RFC1928, March 1996, <https://www.rfc-editor.org/info/rfc1928>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997, <https://www.rfc-editor.org/info/rfc2181>.

[RFC3225] Conrad, D., "Indicating Resolver Support of DNSSEC", RFC 3225, DOI 10.17487/RFC3225, December 2001, <https://www.rfc-editor.org/info/rfc3225>.

[RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record (RR) Types", RFC 3597, DOI 10.17487/RFC3597, September 2003, <https://www.rfc-editor.org/info/rfc3597>.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, DOI 10.17487/RFC4001, February 2005, <https://www.rfc-editor.org/info/rfc4001>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <https://www.rfc-editor.org/info/rfc5952>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <https://www.rfc-editor.org/info/rfc6066>.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. Van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <https://www.rfc-editor.org/info/rfc6147>.

[RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, DOI 10.17487/RFC7050, November 2013, <https://www.rfc-editor.org/info/rfc7050>.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <https://www.rfc-editor.org/info/rfc7231>.

[RFC7595] Thaler, D., Ed., Hansen, T., and T. Hardie, "Guidelines and Registration Procedures for URI Schemes", BCP 35, RFC 7595, DOI 10.17487/RFC7595, June 2015, <https://www.rfc-editor.org/info/rfc7595>.

[RFC7871] Contavalli, C., van der Gaast, W., Lawrence, D., and W. Kumari, "Client Subnet in DNS Queries", RFC 7871, DOI 10.17487/RFC7871, May 2016, <https://www.rfc-editor.org/info/rfc7871>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[WebSocket] Fette, I. and A. Melnikov, "The WebSocket Protocol", RFC 6455, DOI 10.17487/RFC6455, December 2011, <https://www.rfc-editor.org/info/rfc6455>.

15.2. Дополнительная литература

[AltSvc] Nottingham, M., McManus, P., and J. Reschke, "HTTP Alternative Services", RFC 7838, DOI 10.17487/RFC7838, April 2016, <https://www.rfc-editor.org/info/rfc7838>.

[ANAME-DNS-RR] Finch, T., Hunt, E., van Dijk, P., Eden, A., and W. Mekking, "Address-specific DNS aliases (ANAME)", Work in Progress, Internet-Draft, draft-ietf-dnsop-aname-04, 8 July 2019, <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-aname-04>.

[BIND-CHECK-NAMES] Internet Systems Consortium, "BIND v9.19.11 Configuration Reference: "check-names"", September 2023, <https://bind9.readthedocs.io/en/v9.19.11/reference.html#namedconf-statement-check-names>.

[DNAME] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, DOI 10.17487/RFC6672, June 2012, <https://www.rfc-editor.org/info/rfc6672>.

[DNSTerm] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

[ECH] Rescorla, E., Oku, K., Sullivan, N., and C. A. Wood, "TLS Encrypted Client Hello", Work in Progress, Internet-Draft, draft-ietf-tls-esni-17, 9 October 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-esni-17>.

[FETCH] WHATWG, "Fetch Living Standard", October 2023, <https://fetch.spec.whatwg.org/>.

[FETCH-WEBSOCKETS] WHATWG, "WebSockets Living Standard", September 2023, <https://websockets.spec.whatwg.org/>.

[HSTS] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, DOI 10.17487/RFC6797, November 2012, <https://www.rfc-editor.org/info/rfc6797>.

[HTTP-DNS-RR] Bellis, R., "A DNS Resource Record for HTTP", Work in Progress, Internet-Draft, draft-bellis-dnsop-http-record-00, 3 November 2018, <https://datatracker.ietf.org/doc/html/draft-bellis-dnsop-http-record-00>.

[HTTP/3] Bishop, M., Ed., "HTTP/3", RFC 9114, DOI 10.17487/RFC9114, June 2022, <https://www.rfc-editor.org/info/rfc9114>.

[RFC1912] Barr, D., "Common DNS Operational and Configuration Errors", RFC 1912, DOI 10.17487/RFC1912, February 1996, <https://www.rfc-editor.org/info/rfc1912>.

[RFC6454] Barth, A., "The Web Origin Concept", RFC 6454, DOI 10.17487/RFC6454, December 2011, <https://www.rfc-editor.org/info/rfc6454>.

[SRV] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <https://www.rfc-editor.org/info/rfc2782>.

[URI] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

Приложение A. Декодирование текста в файлах зон

Файлы DNS могут представлять произвольные последовательности октетов в виде текста ASCII с применением различных разделителей и кодирования, как указано в параграфе 5.1 [RFC1035]. Ниже приведена сводка разрешённых элементов в форме ABNF.

```
; non-special - VCHAR без DQUOTE, ";", "(", ")", "\".
non-special = %x21 / %x23-27 / %x2A-3A / %x3C-5B / %x5D-7E
; non-digit - VCHAR без DIGIT.
non-digit   = %x21-2F / %x3A-7E
; dec-octet - число (0-255) в виде 3 десятичных цифр.
dec-octet   = ( "0" / "1" ) 2DIGIT /
              "2" ( ( %x30-34 DIGIT ) / ( "5" %x30-35 ) )
escaped     = "\" ( non-digit / dec-octet )
```

```
contiguous  = 1*( non-special / escaped )
quoted      = DQUOTE *( contiguous / ( ["\"] WSP ) ) DQUOTE
char-string = contiguous / quoted
```

Алгоритм декодирования позволяет представлять в char-string любую последовательность *OCTET, используя кавычки для группировки (например, при наличии пробелов) и экранирование для представления непечатаемых октетов в виде escape-последовательностей. В этом документе данный алгоритм называется декодированием символьный строк (character-string decoding), поскольку в параграфе 5.1 [RFC1035] он применяется для создания <character-string>. Несмотря на ограничение размера <character-string> 255 октетами, алгоритм декодирования может давать на выходе строки любого размера.

A.1. Декодирование списка через запятые

Для представления списков элементов файла зоны в этой спецификации применяется разделение запятыми. Если разрешённые элементы списка не включают запятых и \, эта задача тривиальна (для простоты пустые элементы не допускаются). Анализатора списков, разделяющего элементы по запятым и запрещающего символ \, вполне достаточно для выполнения всех требований этого документа. Это соответствует простому синтаксису, приведенному ниже.

```
; item-allowed - строка октетов кез запятых и \.
item-allowed           = %x00-2B / %x2D-5B / %x5D-FF
simple-item            = 1*item-allowed
simple-comma-separated = [simple-item *("," simple-item)]
```

Для реализации, разрешающей элементы с запятыми и \, применяется показанный ниже синтаксис экранирования.

```
item            = 1*OCTET
escaped-item    = 1*(item-allowed / "\," / "\\")
comma-separated = [escaped-item *("," escaped-item)]
```

Декодирование списка значений выполняется после декодирования строк символов. Рассмотрим, например, строки SvcParamValue, показанные ниже.

```
"part1,part2,part3\\,part4\\\\"
part1\,\p\a\r\t2\044part3\092,part4\092\\
```

Эти строки эквивалентны и разбор каждой из них будет давать

```
part1,part2,part3\,part4\\
```

Декодирование списка для этого значения будет давать 3 элемента

```
part1
part2
part3,part4\
```

Приложение B. Сводка отображений HTTP

В таблице 3 приведена ненормативная сводка отображений HTTP для SVCB (раздел 9). В будущие отображения протоколов могут включаться аналогичные таблицы.

Таблица 3.

Отображённая схема	https
Другие затрагиваемые схемы	http, wss, ws, другие на основе HTTP
Тип RR	HTTPS (65)
Префикс имени	Нет для порта 443, иначе _$PORT._https
Автоматически обязательные ключи	port, no-default-alpn
Принятые по умолчанию SvcParam	alpn: [http/1.1]
Особое поведение	Переход от HTTP к HTTPS
Ключи, требуемые в записи	Нет

Приложение C. Сравнение с другими вариантами

Типы записей SVCB и HTTPS очень похожи или вызваны некоторыми имеющимися типами записей и предложениями. Одним из недостатков альтернативных вариантов является отсутствие интереса к их реализации со стороны web-клиентов. Есть надежда, что расширяемое решение для множества проблем преодолеет эту инерцию и будет внедрено клиентами.

C.1. Отличия от типа SRV RR

Запись SRV [SRV] может выполнять функцию, похожую на функцию SVCB, информируя клиента о необходимости поиска службы в другом месте. Однако имеются указанные ниже различия.

- Записи SRV обычно обязательны, а SVCB используются опционально с уже имеющимися протоколами.
- Записи SRV не могут указывать клиенту необходимость изменить или обновить протокол, а SVCB может указывать обновление (например, на HTTP/2).
- Записи SRV не могут расширяться, а в SVCB и HTTPS RR могут добавляться новые параметры.
- В записях SRV указывается вес для несбалансированного случайного распределения нагрузки, а SVCB поддерживает только сбалансированное случайное распределение, хотя в будущие SvcParam может быть добавлен вес.

C.2. Отличия от предложенной записи HTTP

В отличие от [HTTP-DNS-RR], этот подход является расширяемым для охвата применения Alt-Svc и Encrypted ClientHello. Оба предложения решают проблему CNAME на вершине зоны для реализующих их клиентов и в обоих сохраняется необходимость продолжать включать адресные записи на вершине зоны для устаревших клиентов.

C.3. Отличия от предложенной записи ANAME

В отличие от [ANAME-DNS-RR], этот подход является расширяемым для охвата применения Alt-Svc и Encrypted ClientHello. Он не требует каких-либо изменений или специальной обработки на полномочных или первичных серверах, кроме возврата дополнительных записей, относящихся к их сфере ответственности.

Оба предложения решают проблему CNAME на вершине зоны для реализующих их клиентов.

Однако в этом предложении SVCB сохраняется необходимость продолжать включать адресные записи на вершине зоны для устаревших клиентов. Если этот стандарт будет успешно внедрён, число таких клиентов будет сокращаться со временем, что должно привести к сокращению эксплуатационных затрат, связанных с обслуживанием пользователей таких клиентов без перенаправления SVCB. Операторы серверов могут легко увидеть, как много трафика поступает на устаревшие конечные точки и удалить адресные записи на вершине зоны, когда наблюдаемый унаследованный снизится до пренебрежимо малого уровня.

C.4. Сравнение с раздельными типами RR для AliasMode и ServiceMode

Абстрактно функции AliasMode и ServiceMode независимы, поэтому может возникнуть соблазны использовать для них разные типы RR. Однако это привело бы к серьёзному снижению производительности, поскольку клиенты не могли бы полагаться на то, что их рекурсивные распознаватели будут следовать псевдонимам SVCB (подобно CNAME). В результате клиентам пришлось бы параллельно делать запросы для обоих типов RR, причём потенциально на каждом этапе цепочки псевдонимов. Рекурсивные распознаватели, реализующие эту спецификацию, при получении запроса ServiceMode будут передавать полномочному серверу DNS запросы ServiceMode и AliasMode. Таким образом, раздельные типы RR будут удваивать (в некоторых случаях утраивать) нагрузку на клиентов и серверы, а реализацию не упростит.

Приложение D. Тестовые векторы

Приведённые здесь тестовые векторы содержат лишь RDATA записей SVCB/HTTPS в формате представления, базовом формате [RFC3597] и формате передачи. В формате передачи используется шестнадцатеричное представление (\xNN) для байтов, не ASCII. Из-за большого размера строки в формате передачи они разбиты на несколько частей.

D.1. AliasMode

```
example.com.   HTTPS   0 foo.example.com.

\# 19 (
00 00                                                 ; приоритет
03 66 6f 6f 07 65 78 61 6d 70 6c 65 03 63 6f 6d 00 ; цель
)

\x00\x00                                              # приоритет
\x03foo\x07example\x03com\x00                         # цель
```

Рисунок 2. AliasMode.

D.2. ServiceMode

```
example.com.   SVCB   1 .

\# 3 (
00 01      ; приоритет
00         ; цель (метка корня)
)

\x00\x01   # приоритет
\x00       # цель (метка корня)
```

Рисунок 3. TargetName - это . (корень).

```
example.com.   SVCB   16 foo.example.com. port=53

\# 25 (
00 10                                                 ; приоритет
03 66 6f 6f 07 65 78 61 6d 70 6c 65 03 63 6f 6d 00 ; цель
00 03                                                 ; ключ 3
00 02                                                 ; размер 2
00 35                                                 ; значение
)

\x00\x10                                              # приоритет
\x03foo\x07example\x03com\x00                         # цель
\x00\x03                                              # ключ 3
\x00\x02                                              # размер 2
\x00\x35                                              # значение
```

Рисунок 4. Задание порта.

```
example.com.   SVCB   1 foo.example.com. key667=hello

\# 28 (
00 01                                                 ; приоритет
03 66 6f 6f 07 65 78 61 6d 70 6c 65 03 63 6f 6d 00 ; цель
02 9b                                                 ; ключ 667
00 05                                                 ; размер 5
```

```
68 65 6c 6c 6f                                          ; значение
)

\x00\x01                                                # приоритет
\x03foo\x07example\x03com\x00                           # цель
\x02\x9b                                                # ключ 667
\x00\x05                                                # размер 5
hello                                                   # значение
```

Рисунок 5. Базовый ключ и значение без кавычек.

```
example.com.    SVCB    1 foo.example.com. key667="hello\210qoo"

\# 32 (
00 01                                                   ; приоритет
03 66 6f 6f 07 65 78 61 6d 70 6c 65 03 63 6f 6d 00      ; цель
02 9b                                                   ; ключ 667
00 09                                                   ; размер 9
68 65 6c 6c 6f d2 71 6f 6f                              ; значение
)

\x00\x01                                                # приоритет
\x03foo\x07example\x03com\x00                           # цель
\x02\x9b                                                # ключ 667
\x00\x09                                                # размер 9
hello\xd2qoo                                            # значение
```

Рисунок 6. Базовый ключ и значение в кавычках с десятичным экранированием.

```
example.com.    SVCB    1 foo.example.com. (
                        ipv6hint="2001:db8::1,2001:db8::53:1"
                        )

\# 55 (
00 01                                                   ; приоритет
03 66 6f 6f 07 65 78 61 6d 70 6c 65 03 63 6f 6d 00      ; цель
00 06                                                   ; ключ 6
00 20                                                   ; размер 32
20 01 0d b8 00 00 00 00 00 00 00 00 00 00 00 01         ; первый адрес
20 01 0d b8 00 00 00 00 00 00 00 00 00 53 00 01         ; второй адрес
)

\x00\x01                                                # приоритет
\x03foo\x07example\x03com\x00                           # цель
\x00\x06                                                # ключ 6
\x00\x20                                                # размер 32
\x20\x01\x0d\xb8\x00\x00\x00\x00
     \x00\x00\x00\x00\x00\x00\x00\x01                   # первый адрес
\x20\x01\x0d\xb8\x00\x00\x00\x00
     \x00\x00\x00\x00\x00\x53\x00\x01                   # второй адрес
```

Рисунок 7. Две подсказки IPv6 в кавычках.

```
example.com.    SVCB    1 example.com. (
                          ipv6hint="2001:db8:122:344::192.0.2.33"
                          )
\# 35 (
00 01                                                   ; приоритет
07 65 78 61 6d 70 6c 65 03 63 6f 6d 00                  ; цель
00 06                                                   ; ключ 6
00 10                                                   ; размер 16
20 01 0d b8 01 22 03 44 00 00 00 00 c0 00 02 21         ; адрес
)

\x00\x01                                                # приоритет
\x07example\x03com\x00                                  # цель
\x00\x06                                                # ключ 6
\x00\x10                                                # размер 16
\x20\x01\x0d\xb8\x01\x22\x03\x44
     \x00\x00\x00\x00\xc0\x00\x02\x21                   # адрес
```

Рисунок 8. Подсказка IPv6 с использованием синтаксиса Embedded IPv4.

```
example.com.    SVCB    16 foo.example.org. (
                        alpn=h2,h3-19 mandatory=ipv4hint,alpn
                        ipv4hint=192.0.2.1
                        )

\# 48 (
00 10                                                   ; приоритет
03 66 6f 6f 07 65 78 61 6d 70 6c 65 03 6f 72 67 00      ; цель
00 00                                                   ; ключ 0
00 04                                                   ; размер параметра 4
00 01                                                   ; значение: key 1
00 04                                                   ; значение: key 4
00 01                                                   ; ключ 1
00 09                                                   ; размер параметра 9
02                                                      ; размер alpn 2
```

```
68 32                                           ; значение alpn
05                                              ; размер alpn 5
68 33 2d 31 39                                  ; значение alpn
00 04                                           ; ключ 4
00 04                                           ; размер параметра 4
c0 00 02 01                                     ; значение параметра
)

\x00\x10                                        # приоритет
\x03foo\x07example\x03org\x00                   # цель
\x00\x00                                        # ключ 0
\x00\x04                                        # размер параметра 4
\x00\x01                                        # значение: key 1
\x00\x04                                        # значение: key 4
\x00\x01                                        # ключ 1
\x00\x09                                        # размер параметра 9
\x02                                            # размер alpn 2
h2                                              # значение alpn
\x05                                            # размер alpn 5
h3-19                                           # значение alpn
\x00\x04                                        # ключ 4
\x00\x04                                        # размер параметра 4
\xc0\x00\x02\x01                                # значение параметра
```

Рисунок 9. SvcParamKey размещаются произвольно в формате представления, но упорядочены в формате передачи.

```
example.com.    SVCB    16 foo.example.org. alpn="f\\\\oo\\,bar,h2"
example.com.    SVCB    16 foo.example.org. alpn=f\\\092oo\092,bar,h2

\# 35 (
00 10                                           ; приоритет
03 66 6f 6f 07 65 78 61 6d 70 6c 65 03 6f 72 67 00 ; цель
00 01                                           ; ключ 1
00 0c                                           ; размер параметра 12
08                                              ; размер alpn 8
66 5c 6f 6f 2c 62 61 72                         ; значение alpn
02                                              ; размер alpn 2
68 32                                           ; значение alpn
)

\x00\x10                                        # приоритет
\x03foo\x07example\x03org\x00                   # цель
\x00\x01                                        # ключ 1
\x00\x0c                                        # размер параметра 12
\x08                                            # размер alpn 8
f\oo,bar                                        # значение alpn
\x02                                            # размер alpn 2
h2                                              # значение alpn
```

Рисунок 10. Значение alpn с экранированными запятыми и \ в двух форматах представления.

D.3. Отказы

Здесь приведены тестовые векторы, не совместимые с этим документом, и указаны причины несоответствия.

```
example.com.    SVCB    1 foo.example.com. key123=abc key123=def
```

Рисунок 11. Несколько экземпляров одного SvcParamKey.

```
example.com.    SVCB    1 foo.example.com. mandatory
example.com.    SVCB    1 foo.example.com. alpn
example.com.    SVCB    1 foo.example.com. port
example.com.    SVCB    1 foo.example.com. ipv4hint
example.com.    SVCB    1 foo.example.com. ipv6hint
```

Рисунок 12. Отсутствуют SvcParamValue, которые должны быть непустыми.

```
example.com.    SVCB    1 foo.example.com. no-default-alpn=abc
```

Рисунок 13. Значение no-default-alpn SvcParamKey должно быть пустым.

```
example.com.    SVCB    1 foo.example.com. mandatory=key123
```

Рисунок 14. Отсутствует обязательный SvcParam.

```
example.com.    SVCB    1 foo.example.com. mandatory=mandatory
```

Рисунок 15. Недопустимо включать SvcParamKey mandatory в список обязательных.

```
example.com.    SVCB    1 foo.example.com. (
                        mandatory=key123,key123 key123=abc
                        )
```

Рисунок 16. Несколько экземпляров одного SvcParamKey в списке обязательных.

Благодарности и предложения

За прошедшие годы участники IETF предложили широкий спектр решений проблемы «CNAME на вершине зоны», включая [HTTP-DNS-RR], [ANAME-DNS-RR] и др. Авторы выражают благодарность за их работу по прояснению проблемы и определению перспективных стратегий её решения, часть которых отражена в этом документе.

Спасибо Ian Swett, Ralf Weber, Jon Reed, Martin Thomson, Lucas Pardue, Ilari Liusvaara, Tim Wicinski, Tommy Pauly, Chris Wood, David Benjamin, Mark Andrews, Emily Stark, Eric Orth, Kyle Rose, Craig Taylor, Dan McArdle, Brian Dickson, Willem Toorop, Pieter Lexis, Puneet Sood, Olivier Poitrey, Mashooq Muhaimen, Tom Carpay и многим другим за отзывы и предложения по данному документу.

Адреса авторов

Ben Schwartz
Meta Platforms, Inc.
Email: ietf@bemasc.net

Mike Bishop
Akamai Technologies
Email: mbishop@evequefou.be

Erik Nygren
Akamai Technologies
Email: erik+ietf@nygren.org

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru