Network Working Group J. Rosenberg, Ed.
Request for Comments: 4367 IAB
Category: Informational February 2006

Что в имени тебе моем? Ложные представления о доменных именах

What's in a Name: False Assumptions about DNS Names

Аннотация

Система доменных имён DNS[1] обеспечивает важный для работы Internet сервис, отображая структурированные имена на различные типы данных (обычно адреса IP). Эти имена появляются в адресах электронной почты, идентификаторах ресурсов (URI[2]) и других идентификаторах прикладного уровня, которые часто представляются человеку. По этой причине существуют сильные притязания на приобретение имен, значимых для человека благодаря их связи с зарегистрированными торговыми знаками, именами компаний, типами служб и т. п. Такая тенденция таит в себе опасность – люди и автоматизированные системы, использующие такие имена, будут ассоциировать ту или иную семантику с некоторыми именами и, таким образом, делать предположения о том что сервис связан или может быть связан с хостами, которые ассоциируются с именами. Такие допущения часто являются ошибочными, приводя к путанице и возникновению проблем. В данном документе проводится детальное рассмотрение этого вопроса и даются рекомендации по решению проблем.

Оглавление

1. Введение

Система доменных имен DNS [1] обеспечивает важный для работы Internet сервис, отображая структурированные имена на различные типы данных. Наиболее часто эта система используется для получения IP-адреса хоста, связанного с именем этого хоста [2] [1] [3]. Однако эта система может использоваться и для получения другой информации – предположения могут быть сделаны почти обо всем, включая географическое положение [4].

Доменные имена чаще всего применяются в идентификаторах, используемых прикладными протоколами. К наиболее распространенным применениям относятся адреса электронной почты, идентификаторы ресурсов URI (такие, как HTTP URL [5]), RTSP[3] URL [6] и SIP URI [7]. Эти идентификаторы являются уникальными, их указывают на визитных карточках, web-страницах, уличной рекламе и т. п. По этой причине существуют сильные притязания на приобретение имен, значимых для человека благодаря их связи с зарегистрированными торговыми знаками, именами компаний, типами служб и т. п. Такие идентификаторы могут использоваться в бизнесе, включая продвижение торговых марок (brand), рекламу и т. п.

Люди часто делают предположения о типе сервиса, который обеспечивается или может быть обеспечен хостом, связанным с определенным именем, на основе своих представлений и ожиданий, основанных на толковании этого

[1]Domain Name System
[2]Uniform Resource Identifier – типовой идентификатор ресурса
[3]Real Time Streaming Protocol – потоковый протокол в реальном масштабе времени.

имени. Это приводит к попыткам организаций регистрировать доменные имена на основе предполагаемых пользовательских ожиданий. Примером этого могут служить различные предложения для имен доменов верхнего уровня (TLD[1]), которые могут быть связаны с информацией «только для взрослых» (adult content) [8], запросы на создание TLD, связанных с мобильными устройствами и службами, и даже «разведение кроликов» (phishing attacks).

Когда такие предположения переносятся в поведение автоматизированных систем (таких, как клиенты или серверы прикладного уровня) в результате выбора разработчиков, управляющих директив или политики владельца домена, в системах могут возникать различные сбои. В этом документе описано множество типичных способов такого сопоставления имен, связанных с этим проблем, последствий ошибок и рекомендаций по предотвращению проблем.

В главе 4 описаны некоторые возможные предположения, которые клиенты, серверы и люди могут делать в части толкования доменных имен. В этом контексте термин «предположение» (assumption) означает поведение, ожидаемое при обращении к сервису с данным именем даже если такое поведение явно не задано спецификациями протокола. Зачастую такие предположения включают игнорирование части спецификации на основе допущения, что клиенты и серверы используются в среде, требования которой более жестки, нежели требования спецификации. В главе 5 приводится обзор некоторых последствий таких ошибочных предположений. В общем случае такие последствия могут включать множество различных проблем взаимодействия, возникновение сложностей при работе пользователей и системные отказы. Глава 6 посвящена обсуждению причин, по которым такие предположения могут быть ошибочными изначально или становиться таковыми с течением времени. Чаще всего такие предположения становятся ошибочными в результате неожиданного изменения среды с течением времени, когда правильные допущения становятся ложными. Иногда предположения становятся ошибочными в результате того, что они были основаны на участии конкретного множества клиентов и серверов, а с течением времени появляются новые участники.

В главе 7 содержатся некоторые рекомендации. Эти рекомендации включают в себя инженерный опыт, накопленный за десятилетия разработки протоколов Internet. К таким рекомендациям относятся:

- строгое следование спецификациям;
- использование возможностей согласования, обеспечиваемых протоколом;
- либеральная политика по отношению к другим (прием данных) и консервативное отношение к себе (передача данных) [18].

В любом случае автоматизированным системам не следует изменять поведение протокола на основе доменного имени хоста или компонент такого имени.

2. Аудитория

Этот документ предназначен для нескольких категорий читателей. Во-первых он адресован разработчикам программ, которые в конечном счете и создают приложения, делающие ложные предположения. Приведенные здесь рекомендации служат для подкрепления рекомендаций, содержащихся в спецификациях и известных разработчикам, но часто забываемых ими.

Документ адресован также специалистам, разрабатывающим требования к приложениям, которые ведут в возникновению ложных допущений. Для них этот документ подчеркнет важность того, что не следует принимать скороспелых решений по вопросам применимости проекта.

Кроме того, документ предназначен администраторам и другим лицам, определяющим политику доменов. Для них документ показывает риск создания политики домена, которая может оказывать влияние на работу приложений внутри этого домена.

3. Модель использования DNS

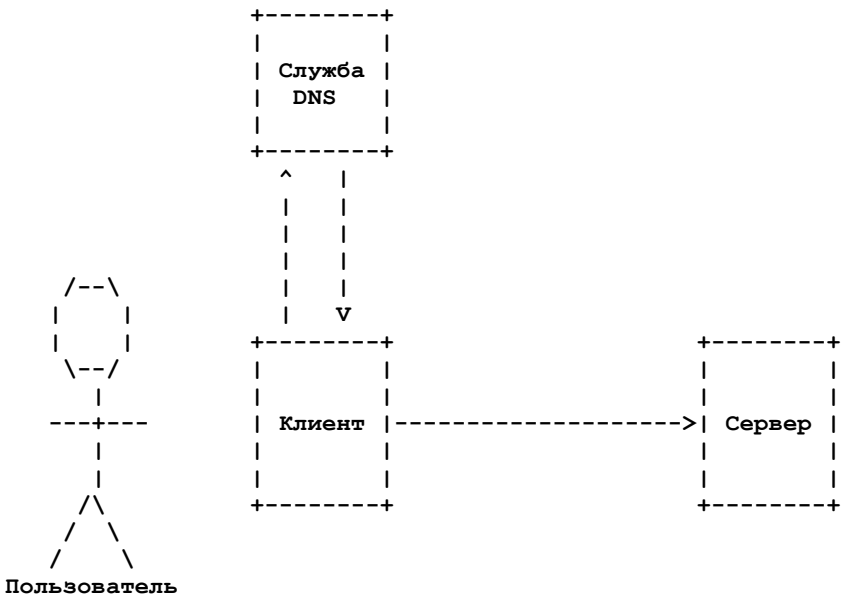

Рисунок 1.

На рисунке 1 показана простая концептуальная модель использования DNS приложениями. Пользователь приложения знает идентификатор для интересующей информации или службы. Этот идентификатор зачастую представляет собой URL или URI, содержащий доменное имя. Пользователь вводит идентификатор в клиентском приложении (например,

[1]Top-Level Domain.

набирая URL в строке ввода окна браузера). Клиент представляет собой автоматическую программно-аппаратную систему, которая контактирует с соответствующим сервером для того, чтобы обслужить запрос пользователя. Для того, чтобы сделать это, клиент обращается к серверу DNS для преобразования доменного имени из полученного идентификатора в адрес IP. После этого клиент обращается к серверу по полученному адресу. Эта простая модель применима к таким протоколам, как HTTP [5], SIP [7], RTSP [6], SMTP [9].

При рассмотрении модели становится очевидным, что в системе присутствует три компоненты, которые могут использовать ложные предположения о предоставляемых сервером услугах. Человек может формировать свои ожидания в части содержания предоставляемых услуг на своем толковании имени хоста, обеспечивающего сервис. Сервер может предполагать, что подключающийся к нему клиент поддерживает протоколы, которых тот реально не поддерживает, способен воспринимать информацию, которую клиент реально не понимает, или поддерживает возможности, которых нет у реального клиента. Аналогично, клиент может предполагать на сервере поддержку протоколов, содержимого и возможностей, которые не поддерживаются в реальности. Более того, приложения могут включать множество людей, клиентов и серверов, каждый из которых независимо от других может делать свои ложные предположения.

4. Возможные допущения

Каждый из трех элементов модели может делать множество различных ложных предположений.

4.1. Допущения пользователей

Множество возможных предположений пользователя практически не ограничено. Пользователи могут предполагать, что идентификатор HTTP URL, имеющий сходство с именем компании, приведет их на сервер этой компании. Они могут предполагать, что почтовый адрес из домена верхнего уровня .gov реально указывает на государственного служащего. Пользователи могут предполагать, что документы на web-сервере с домене TLD, выделенном для информации “только для взрослых” (например, .sex), действительно содержат информацию такого рода. Такие предположения неизбежны, все они могут оказаться ложными и не являются основной темой данного документа.

4.2. Допущения клиентов

Несмотря на то, что клиент является “автоматическим”, он может делать некоторые предположения, присущие человеку. Например, многие клиенты предполагают, что любой хост, имя которого начинается с www, является web-сервером, хотя такое предположение явно может быть ложным.

Кроме того, клиент связывает себя с протоколами, требуемыми для обмена данными с сервером. В результате он делает предположения о работе протоколов для связи с сервером. Эти предположения проявляются в реализации, когда игнорируются стандартные методы согласования, определенные протоколом, а взамен используется тот или иной набор правил, заложенных в программу и делающий свое заключение о согласовании параметров. Результатом этого зачастую является потеря интероперабельности, снижение уровня надежности или обретение пользователями негативного опыта.

Алгоритм аутентификации: Несмотря на поддержку протоколом множества методов аутентификации, клиент может предположить, что сервер всегда поддерживает только один метод, который, согласно протоколу, является необязательным. Например, клиент SIP, контактирующий с сервером SIP в домене, который используется для идентификации мобильных устройств (например, www.example.cellular) может предположить, что сервер необязательный поддерживает метод AKA[1] [10] на основании доменного имени, используемого для доступа к серверу. Другим примером может служить web-клиент, предполагающий, что сервер с именем https.example.com поддерживает защищенный протокол HTTP over TLS[2] [16].

Форматы данных: Несмотря на поддержку протоколом множества форматов передачи данных клиенту от сервера, клиент может предположить использование какого-то одного метода взамен применения меток содержимого и возможностей согласования, обеспечиваемых нижележащим протоколом. Например, клиент RTSP может предположить, что все аудио-данные, получаемые с сайта media.example.cellular, используют узкополосное кодирование. В качестве другого примера можно рассмотреть почтового клиента, который полагает, что почтовый сервер с именем mail.example.cellular поддерживает только текстовый формат сообщений вместо проверки заголовков MIME [11] в сообщении для определения реального типа.

Расширения протокола: Клиент может попытаться работать с сервером, используя необязательные расширения протокола. Однако при этом вместо реализации требуемой логики выбора расширений (fallback logic), клиент может сделать ложное допущение о поддержке сервером того или иного расширения. Примером может служить клиент SIP, который требует гарантированных откликов на свои запросы (RFC 3262 [17]), предполагая, что это расширение поддерживается сервером с именем sip.example.telecom. Более того, клиент не будет реализовать возможность отказа от дополнительных расширений (fallback behavior), определенную в RFC 3262, поскольку предполагает, что все серверы, с которыми он будет взаимодействовать, находятся в том же домене и, следовательно, поддерживают это расширение. Однако, если это предположение окажется ошибочным, клиент не сможет сделать ни одного телефонного звонка.

Языки: Клиент может поддерживать средства обработки текста в зависимости от языка этого текста. Вместо определения языка на основе маркеров в сообщении от сервера клиент может делать предположения о языке на основе доменного имени. Такие предположения зачастую оказываются ложными. Например, клиент может предположить, что текст web-страницы с сервера в национальном (ccTLD) домене .de содержат текст на немецком языке и попытаться перевести этот текст на финский язык. Результат явно ошеломит пользователя, если реальный текст будет написан на французском языке. К несчастью описанное поведение клиентов иногда бывает вызвано тем, что сервер не указывает корректно язык документов, полагая, что это необязательно. Этот пример показывает, как ложные допущения могут создавать порочный круг.

[1] Authentication and Key Agreement – аутентификация и согласование ключей.
[2] Transport Layer Security – защита транспортного уровня.

4.3. Допущения серверов

Сервер, подобно клиенту, представляет собой автоматическую систему. Будем рассматривать для примера сервер www.company.cellular. Этот сервер может предполагать, что все подключающиеся к нему клиенты поддерживают конкретные возможности вместо того, чтобы использовать протоколы нижележащих уровней для определения возможностей клиентов. Допущения такого типа перечислены ниже.

Алгоритм аутентификации: Сервер может предполагать, что клиент поддерживает конкретный необязательный метод аутентификации и, следовательно, не поддерживает обязательный метод аутентификации.

Язык: Сервер может использовать тот или иной язык для представления документов на основе предположения, что обращающиеся к этому домену клиенты понимают этот язык, или предположения о том, что клиент с определенным адресом IP говорит на соответствующем языке.

Форматы данных: Сервер может предполагать, что клиент использует определенный набор типов MIME и способен передавать только такие типы. При генерации содержимого в протокольном отклике сервер игнорирует заголовки согласования содержимого, которые могли присутствовать в запросе. Например, сервер может игнорировать тег Accept HTTP и передавать изображение в том или ином специфическом формате.

Расширения протокола: Сервер может предполагать, что клиент поддерживает то или иное необязательное расширение протокола и не поддерживать согласование, которое необходимо, если клиент не поддерживает это расширение.

Характеристики клиента: Сервер может делать некоторые предположения о физических характеристиках своих клиентов (объем памяти, производительность процессора, размер экрана, глубина цвета, указательное устройство и т. п.). Основываясь на таких допущениях, сервер может выбирать свое поведение при обработке запроса. Например, web-сервер может всегда предполагать, что клиент подключен через сотовый телефон и, следовательно, возвращать содержимое с минимальным количеством графики и иными особенностями, используемые в подобных случаях.

5. Последствия ложных допущений

В результате ложных допущений серверов, клиентов и пользователей может возникать множество негативных последствий. Ниже перечисляются некоторые из них.

Проблемы взаимодействия: В некоторых случаях клиент или сервер предполагает определенный тип работы протокола и это предположение является ошибочным. В результате такой ошибки клиент и сервер не могут нормально взаимодействовать и пользователь получает сообщение о той или иной ошибке. Проблемы этого типа носят постоянный характер и при повторных попытках обращения клиента ошибка будет повторяться. Решением проблемы может быть лишь замена клиентской или серверной программы.

Системные сбои: В отдельных случаях клиент или сервер некорректно интерпретирует протокольные операции и это открывает дефекты реализации. Дефект приводит к постоянному (до сброса с участием человека) или временному полному отказу системы или выходу из строя отдельных компонент. Если отказ возникает на сервере, проблема коснется не только клиента, вызвавшего этот отказ, но и всех прочих клиентов, которые будут обращаться к серверу. Например, если web-сервер предполагает, что информация, полученная им от клиента (например, изображение с цифровой камеры) относится к определенному типу, и всегда передает изображение в кодек для декомпрессии прежде, чем записать его, работа кодека может завершиться отказом при сжатии исходного изображения с использованием неподдерживаемого кодеком формата. Такая проблема может возникать даже при корректном значении Content-Type, если сжатый битовый поток содержит ошибки. Ложные допущения увеличивают вероятность отказа.

Негативный опыт пользователей: В таких случаях клиент и сервер могут взаимодействовать, но пользователь приобретает негативный опыт. Например, если клиент на ПК подключается к web-серверу, предназначенному для работы с мобильными устройствами, содержимое полученных документов может оказаться неудобочитаемым на экране ПК, Другим примером может служить обращение клиента к потоковому сервису, при котором информация передается в низкоскоростном варианте, хотя кодек клиента может поддерживать более высокое качество. Еще одним примером является ситуация, когда пользователь желает обращаться к сервису с ПК и мобильного устройства, используя общую адресную книгу (эта книга доступна с разных устройств). В таких случаях пользователю нужно создавать две записи в адресной книге и использовать при обращении к серверу запись, соответствующую выбранному для работы с сервером устройству. Такие ситуации оставляют у пользователя негативные впечатления.

Снижение уровня безопасности: В таких случаях используется менее надежный механизм обеспечения безопасности, нежели следует использовать. Например, сервер в домене может предположить, что он работает только с клиентами, поддерживающими ограниченный набор механизмов защиты, несмотря на то, что клиенты реально могут поддерживать наиболее мощные средства обеспечения безопасности.

6. Причины ложных допущений

Предположения о работе протоколов, которые клиенты и серверы делают при контактах с определенным доменом, являются весьма зыбкими и могут оказаться ложными в силу различных причин. На серверной стороне многие допущения основываются на том, что данное доменное имя используется лишь ограниченным множеством клиентов. Если держатель доменного имени делает какие-то предположения о клиентах и полагает, что только такие клиенты используют сервис, он может настроить сервер на работу именно с такими клиентами. Однако такие предположения зачастую оказываются ложными, как будет показано ниже.

На клиентской стороне предположения основываются на том, что сервер, расположенный в определенном домене, будет предоставлять специфический тип сервиса. Передача полномочий или иные изменения, обсуждаемые ниже, могут делать такие допущения ложными.

6.1. Эволюция

Internet и устройства, используемые для доступа в сеть постоянно изменяются и, зачастую, весьма быстро. К сожалению существует тенденция строить «здесь и сейчас», не задумываясь о будущем. Многие из перечисленных

выше предположений основываются на характеристиках сегодняшних клиентов и серверов. Поддержка специфических протоколов, методов аутентификации или содержимого основывается на сегодняшних стандартах и устройствах. Хотя такой подход является правильным для большинства случаев, это не может происходить всегда. Прекрасным примером являются мобильные устройства. Сервер, обслуживающий домен, к которому обращаются такие устройства, может делать предположения о протоколах, расширениях, размере экранов, механизмах защиты или производительности процессоров таких устройств. Однако все эти характеристики могут и будут меняться с течением времени.

Такие изменения обычно происходят эволюционным путем. В результате принятые допущения остаются корректными в некоторых случаях, но это бывает не всегда. Достаточно сложно исправлять подобные системы, поскольку для этого обычно требуется определение сервером типа подключенного клиента и поддерживаемых им возможностей. Если при создании и развертывании системы не были заложены функции согласования возможностей, определение характеристик клиентов становится очень сложной задачей. На практике исправление таких систем зачастую требует добавления поддержки согласования возможностей. Если же такие функции заложены изначально, это позволит избежать проблем в будущем.

6.2. Распространение информации

Серверы также делают предположения, основанные на убеждении в том, что к ним будут обращаться только клиенты определенного типа, настроенные для работы с этим доменным именем. В сущности, сервер предполагает, что его знает и использует лишь ограниченная группа (community), а для всех прочих его просто не существует.

Проблема состоит в том, что это допущение обычно является ложным. Сеть Internet является глобальной, так же, как служба доменных имен DNS. Не существует технических барьеров, отделяющих членов группы (community) от всех остальных. С учетом распространения информации по сети Internet очевидно, что к серверу время от времени будут обращаться клиенты, которые не входят в предполагаемую группу. Повсеместное присутствие доменных имен в различных форматах URI, вкупе с простотой распространения URI делает процесс распространения информации о доменных именах чрезвычайно быстрым. Более того, глобальный характер DNS с единственным корнем системы имен [12] делает возможным для клиентов, не включенных в группу поиск, нахождение и использование таких “специальных” доменных имен.

Такое распространение информации является сильной стороной архитектуры Internet, а не ее недостатком. Таким путем обеспечивается глобальный доступ к сервису с помощью любого клиента, подключенного к сети Internet. В конечном итоге это обеспечивает быстрый рост числа пользователей любого конкретного сервиса.

6.3. Передача полномочий

Клиенты и серверы делают предположения о доменах в силу представления о том, что что существует некий централизованный контроль, который может служить основой для таких предположений. Однако система DNS является не централизованной, а распределенной. Если домен не передает свои полномочия в субдомены, все записи этого домена хранятся в одной зоне и может использоваться централизованная политика управления работой домена. Однако в достаточно больших доменах часть полномочий зачастую передается в субдомены и это осложняет централизованный контроль природы услуг, обеспечиваемых каждым из субдоменов.

Использование доменных имен с удобной для человека семантикой ведет к регистрации множества доменов, в которых может быть реализован тот или иной конкретный сервис. Например, сервис-провайдер по имени example может зарегистрировать и развернуть свои службы в доменах example.com, example.net и, в общем случае в example.foo, где foo может быть любым корректным TLD. Это, подобно передаче полномочий, приводит к росту числа доменов и осложняет централизованный контроль.

Известно множество примеров успешной реализации централизованного управления для многоуровневой структуры субдоменов. Однако это требует значительных усилий и участия человека в создании процессов и процедур. Автоматическая проверка соответствия правилам весьма затруднительна и не существует способа автоматической проверки выполнения множества правил.

Более простым вариантом обеспечения централизованного управления будет уповать на то, что централизованные правила будут выполняться и потом ждать жалоб на их нарушение или время от времени проводить проверки выполнения правил (аудит). Этот вариант связан с множеством проблем.

Более детальное рассмотрение некорректности допущений в результате передачи полномочий приводится в параграфе 4.1.3 документа [8] и параграфе 3.3 документа [20].

Как результат слабого контроля за соблюдением единой политики при передаче полномочий является то, что если пользователь или клиент может предположить некоторые свойства, связанные с TLD (например, .wifi), такие свойства могут теряться при передаче полномочий и не поддерживаться конкретным сервером в данном домене верхнего уровня. Например, в store.chain.company.provider.wifi может быть 4 уровня передачи полномочий от .wifi и достаточно очевидно, что при отсутствии должных усилий со стороны держателя .wifi, некоторые свойства могут отсутствовать на нижних уровнях. Причинами отсутствия этих свойств могут быть человеческие ошибки или осознанный отказ от соответствия общей политике.

6.4. Мобильность

Одним из основных назначений имени хоста является индикация его постоянного существования. Клиент может изменить адрес IP, сохраняя постоянный идентификатор, используемый другими хостами для доступа к нему. Поскольку имя связано с хостом, оно может перемещаться с ним не только при смене адреса IP, но и в результате изменения провайдера или технологии подключения к сети. По этой причине предположения о хостах, сделанные на основе предполагаемой сети доступа, соответствующей имени этого хоста, имеют тенденцию становиться ложными с течением времени. Например, ПК может подключаться к провайдеру широкополосного доступа и через динамический сервер DNS получать имя в домене провайдера. Однако не следует полагать, что любой хост в этой сети подключен с использованием широкополосного канала – пользователь может подключить свой ПК через узкополосный канал беспроводного доступа и сохранить свое доменное имя.

6.5. Человеческие ошибки

Человеческие ошибки могут проявляться в любой системе и рассматриваемые случаи не являются исключением. Существует множество ошибок, связанный с обсуждаемой проблемой.

Реализация клиента может предполагать, что существование записи DNS SRV для определенного протокола в определенном домене, показывающее что сервис доступен через некий порт, говорит о том, что сервис фактически развернут здесь. Это предположение может оказаться ложным в результате того, что системный администратор забыл обновить запись SRV для реального сервиса. Другим примером может служить предположение клиента о том, что правила некого домена применимы ко всем его субдоменам. Однако администратор может просто забыть применить правила к серверам, работающим в одном из таких субдоменов.

7. Рекомендации

Из приведенного выше рассмотрения различных проблем с очевидностью следует, что клиенты, серверы и пользователи не должны делать предположений о природе сервиса, обеспечиваемого или используемого доменом. Ниже приводятся более конкретные рекомендации.

Соответствие спецификациям. Когда спецификация определяет обязательные базовые процедуры и форматы, их следует реализовать даже в тех случаях, когда ожидается использование дополнительных процедур. Например, если спецификация требует использования определенного базового метода аутентификации, но позволяет использовать по согласованию другие методы, разработчикам следует реализовать базовый алгоритм, даже в тех случаях, когда другие алгоритмы используются более часто. Проще говоря, поведение протокола никогда не следует изменять на основе доменного имени хоста.

Согласование возможностей. Многие протоколы поддерживают механизмы согласования возможностей. Например, схема согласования содержимого определена для протоколов, использующих MIME [13] [14] [15]. Протокол SIP позволяет клиентам согласовать тип среды, используемой в multimedia-сеансе, а также параметры протокола. HTTP позволяет клиентам согласовать тип информации, возвращаемой по запросу. Когда такие возможности поддерживаются протоколом, клиентам и серверам следует использовать эти возможности и не делать предположений о поддерживаемых другой стороной возможностях. Разработчикам протоколов следует включать механизмы согласования в тех случаях, когда предполагается возможность изменения способов использования протокола.

Требовательность к себе и великодушие к другим [18]. Эта аксиома протоколов Internet применима и для рассматриваемой проблемы. Разработчикам следует быть готовыми к приему всего, что протокол позволяет передать другой стороне и не ограничиваться приемом лишь того, что хочется получить.

В качестве заключения отметим, что нет необходимости делать какие-либо предположения на основании доменных имен. Вместо этого следует использовать спецификации, обеспечиваемые ими возможности согласования и тогда система будет устойчива к ошибкам и интероперабельна.

8. Примечания к RFC 2219 и RFC 2782

В соответствии с данными здесь определениями предположений, поведение, обусловленное записями DNS, также является основанным на предположении. RFC 2219 [19] определяет общепринятые (well-known) псевдонимы, которые могут использоваться при создании доменных имен, используемых для доступа к общепринятым службам в домене. Этот подход был впоследствии расширен путем определения нового типа записи о ресурсах SRV [2], который используется для указания того или иного типа сервиса, работающего на сервере в данном домене. Оба эти механизма могут быть полезными намеками на сервис, реализованный в домене, но эти намеки могут стать основой для ложных допущений. Однако причины, по которым такие допущения могут быть ложными, отличаются от рассмотренных выше.

Клиент, предполагающий, что хост ftp.example.com является сервером FTP, может ошибаться по той причине, что соглашение об именах RFC 2219 может быть неизвестно владельцу домена или не выполняться им. В соответствии с RFC 2782 запись SRV для того или иного сервиса включается только по желанию администратора домена и клиент, делающий предположения о том, что хост обеспечивает некий сервис, может ошибаться в результате допущенной человеком ошибки. В этом случае ошибочность допущения менее очевидна, но, тем не менее, оно может быть ошибочным.

Единственным способом проверки наличия сервиса на данном хосте является попытка организации соединения с портом, используемым для этого сервиса. Разработчикам следует внимательно относится к тому, чтобы не возникало отказов в тех случаях, когда в записи содержится ложная информация.

9. Вопросы безопасности

Одним из допущений, которые могут принимать клиенты и серверы является предположение о доступности и возможности использования (или обратное предположение) того или иного протокола или механизма защиты. Например, клиент, обращающийся к службе определенного домена может предполагать наличие специфического алгоритма аутентификации или функции хэширования на уровне прикладного протокола. Возможно, что с течением времени этот механизм будет признан недостаточно сильным, что потребует его замены. Возможны и ситуации, когда система начинает использовать незащищенный механизм, но впоследствии переходит к тому или иному варианту защиты. В любом случае предположения о свойствах системы защиты могут приводить к осложнению взаимодействия или, хуже того, предоставлению услуг в незащищенном варианте, несмотря на то, что клиент и сервер поддерживают защищенный режим. Этот тип допущений является принципиально ошибочным даже в тех случаях, когда сами записи защищены с помощью DNSSEC.

10. Благодарности

IAB выражает свою благодарность John Klensin, Keith Moore и Peter Koch за их комментарии к документу.

11. Члены IAB

Членами IAB[1] на момент написания этого документа были:

Bernard Aboba
Loa Andersson
Brian Carpenter
Leslie Daigle
Patrik Faltstrom
Bob Hinden
Kurtis Lindqvist
David Meyer
Pekka Nikander
Eric Rescorla
Pete Resnick
Jonathan Rosenberg

12. Литература

[1] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[2] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[3] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", RFC 3403, October 2002.

[4] Davis, C., Vixie, P., Goodwin, T., and I. Dickinson, "A Means for Expressing Location Information in the Domain Name System", RFC 1876, January 1996.

[5] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[6] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

[7] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[8] Eastlake, D., ".sex Considered Dangerous", RFC 3675, February 2004.

[9] Klensin, J., "Simple Mail Transfer Protocol", RFC 2821, April 2001.

[10] Niemi, A., Arkko, J., and V. Torvinen, "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)", RFC 3310, September 2002.

[11] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[12] Internet Architecture Board, "IAB Technical Comment on the Unique DNS Root", RFC 2826, May 2000.

[13] Klyne, G., "Indicating Media Features for MIME Content", RFC 2912, September 2000.

[14] Klyne, G., "A Syntax for Describing Media Feature Sets", RFC 2533, March 1999.

[15] Klyne, G., "Protocol-independent Content Negotiation Framework", RFC 2703, September 1999.

[16] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[17] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 3262, June 2002.

[18] Braden, R., "Requirements for Internet Hosts – Communication Layers", STD 3, RFC 1122, October 1989.

[19] Hamilton, M. and R. Wright, "Use of DNS Aliases for Network Services", BCP 17, RFC 2219, October 1997.

[20] Faltstrom, P., "Design Choices When Expanding DNS", Work in Progress[2], June 2005.

Адрес автора

Jonathan Rosenberg, Editor
IAB
600 Lanidex Plaza
Parsippany, NJ 07054
US
Phone: +1 973 952-5000
EMail: jdrosen@cisco.com
URI: http://www.jdrosen.net

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

[1]Internet Architecture Board - комитет по архитектуре Internet.
[2]Работа опубликована в RFC 5507. *Прим. перев.*

Полное заявление авторских прав

Интеллектуальная собственность

IETF не принимает какой-либо позиции в отношении действительности или объема каких-либо прав интеллектуальной собственности (Intellectual Property Rights или IPR) или иных прав, которые, как может быть заявлено, относятся к реализации или использованию описанной в этом документе технологии, или степени, в которой любая лицензия, по которой права могут или не могут быть доступны, не заявляется также применение каких-либо усилий для определения таких прав. Сведения о процедурах IETF в отношении прав в документах RFC можно найти в BCP 78 и BCP 79.

Копии раскрытия IPR, предоставленные секретариату IETF, и любые гарантии доступности лицензий, а также результаты попыток получить общую лицензию или право на использование таких прав собственности разработчиками или пользователями этой спецификации, можно получить из сетевого репозитория IETF IPR по ссылке http://www.ietf.org/ipr.

IETF предлагает любой заинтересованной стороне обратить внимание на авторские права, патенты или использование патентов, а также иные права собственности, которые могут потребоваться для реализации этого стандарта. Информацию следует направлять в IETF по адресу ietf-ipr@ietf.org.

Подтверждение

Финансирование функций RFC Editor обеспечивается IETF Administrative Support Activity (IASA).