Internet Engineering Task Force (IETF)
Request for Comments: 7301
Category: Standards Track
ISSN: 2070-1721

S. Friedl
Cisco Systems, Inc.
A. Popov
Microsoft Corp.
A. Langley
Google Inc.
E. Stephan
Orange
July 2014

Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension

Расширение TLS для согласования протокола прикладного уровня

Аннотация

В этом документе описано расширение протокола защиты на транспортном уровне (Transport Layer Security или TLS) для согласования протокола прикладного уровня в процессе согласования TLS. Для экземпляров с поддержкой нескольких протоколов прикладного уровня на одном порту TCP или UDP это расширение позволяет прикладному уровню выбрать протокол для использования в соединении TLS.

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 5741.

Информация о текущем статусе документа, найденных ошибках и способах обратной связи доступна по ссылке http://www.rfc-editor.org/info/rfc7301.

Авторские права

Copyright (c) 2014. Авторские права принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права защищены.

К документу применимы права и ограничения, указанные в BCP 78 и IETF Trust Legal Provisions и относящиеся к документам IETF (http://trustee.ietf.org/license-info), на момент публикации данного документа. Прочтите упомянутые документы внимательно. Фрагменты программного кода, включённые в этот документ, распространяются в соответствии с упрощённой лицензией BSD, как указано в параграфе 4.e документа IETF Trust Legal Provisions, без каких-либо гарантий (как указано в Simplified BSD License).

Оглавление

1. Введение

Протоколы прикладного уровня все чаще инкапсулируются в TLS [RFC5246]. Эта инкапсуляция позволяет приложениям использовать имеющиеся защищённые каналы связи, уже присутствующие на порту 443 почти во всей глобальной инфраструктуре IP.

При поддержке нескольких протоколов приложений не одном порту серверной стороны (например, 443) клиенту и серверу нужно согласовать протокол, применяемый в каждом соединении. Желательно выполнить это согласование без дополнительных круговых обходов (round-trip) между клиентом и сервером, поскольку каждый такой обход снижает качество для конечного пользователя. Кроме того, полезно разрешить выбор сертификатов на основе согласованного протокола приложения.

Этот документ задаёт расширение TLS, позволяющее прикладному уровню согласовать выбор протокола в процессе согласования TLS (handshake). Это было запрошено рабочей группой HTTPbis для решения задачи согласования

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

HTTP/2 ([HTTP2]) через TLS, однако ALPN облегчает согласование произвольных протоколов прикладного уровня. При использовании ALPN клиент передаёт список поддерживаемых прикладных протоколов как часть сообщения TLS ClientHello. Сервер выбирает протокол и передаёт его как часть сообщения TLS ServerHello. Таким образом, можно согласовать протокол прикладного уровня в рамках согласования TLS без добавления круговых обходов через сетьhandshake, а сервер при желании может связать свой сертификат с каждым прикладным протоколом.

2. Уровни требований

Ключевые слова **должно** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с [RFC2119].

3. Согласование протокола прикладного уровня

3.1. Расширение для согласования протокола прикладного уровня

Определён новый тип расширения application_layer_protocol_negotiation(16), который клиент **может** включать в своё сообщение ClientHello" .

```
enum {
    application_layer_protocol_negotiation(16), (65535)
} ExtensionType;
```

В поле extension_data расширения application_layer_protocol_negotiation(16) **нужно** включать список ProtocolNameList.

```
opaque ProtocolName<1..2^8-1>;

struct {
    ProtocolName protocol_name_list<2..2^16-1>
} ProtocolNameList;
```

ProtocolNameList содержит список протоколов, анонсируемых клиентом, в порядке снижения их предпочтительности. Протоколы указываются зарегистрированными IANA не обрабатываемыми непустыми строками байтов, как описано в разделе 6. **Недопустимо** включение пустых строк и отсечка строк байтов.

Сервер, получивший ClientHello с расширением application_layer_protocol_negotiation, **может** вернуть клиенту выбранный протокол. Нераспознанные протоколы сервер будет игнорировать. Клиенту **может** возвращаться новый тип расширения ServerHello application_layer_protocol_negotiation(16) в сообщении ServerHello. Поле extension_data в расширении application_layer_protocol_negotiation(16) структурировано также как описанное выше клиентское поле extension_data, но в ProtocolNameList **должно** содержаться лишь одно значение ProtocolName.

В результате полное согласование с расширением application_layer_protocol_negotiation в сообщениях ClientHello и ServerHello имеет вид, показанный на рисунке 1 (в отличие от указанного в параграфе 7.3 [RFC5246]).

```
Клиент                                  Сервер

ClientHello                 -------->   ServerHello
  (расширение ALPN и                      (расширение ALPN и
   список протоколов)                      выбранный протокол)
                                        Certificate*
                                        ServerKeyExchange*
                                        CertificateRequest*
                            <--------   ServerHelloDone
Certificate*
ClientKeyExchange
CertificateVerify*
[ChangeCipherSpec]
Finished                    -------->
                                        [ChangeCipherSpec]
                            <--------   Finished
Application Data            <------->   Application Data
```

** указывает необязательные или зависящие от ситуации сообщения, передаваемые не всегда.*

Рисунок 1.

Сокращённое согласование с расширением application_layer_protocol_negotiation показано на рисунке 2.

```
Клиент                                  Сервер

ClientHello                 -------->   ServerHello
  (расширение ALPN и                      (расширение ALPN и
   список протоколов)                      выбранный протокол)
                                        [ChangeCipherSpec]
                            <--------   Finished
[ChangeCipherSpec]
Finished                    -------->
Application Data            <------->   Application Data
```

Рисунок 2.

В отличие от многих других расширений TLS, это расширение задаёт свойства не сессии, а соединения. При восстановлении сессии или использовании сеансовых квитанций [RFC5077] предыдущее содержимое этого расширения не имеет значения и учитываются лишь значения, переданные при новом согласовании.

3.2. Выбор протокола

Предполагается, что сервер будет иметь список поддерживаемых им протоколов в порядке предпочтения и выбирать из него 1 поддерживаемый клиентом протокол. Серверу **следует** выбирать наиболее предпочтительный протокол, который поддерживает клиент. Если сервер не поддерживает ни одного из анонсированных клиентом протоколов, ему нужно ответить критическим предупреждением no_application_protocol.

```
enum {
    no_application_protocol(120),
    (255)
} AlertDescription;
```

Протокол, указанный расширением application_layer_protocol_negotiation в ServerHello, **нужно** устанавливать в соединении до пересогласования. Серверу **не следует** указывать один выбранный протокол, а затем применять другой протокол для обмена данными приложения.

4. Конструктивные соображения

Расширение ALPN разработано в соответствии с типичными расширениями протокола TLS. В частности, согласование полностью выполняется в процессе обмена приветственными сообщениями (hello) между клиентом и сервером в соответствии с установленной архитектурой TLS. Расширение ServerHello application_layer_protocol_negotiation должно быть определяющим для соединения (до пересогласования) и передаётся в открытом виде, чтобы позволить элементам сети предоставлять для соединения дифференцированные услуги, когда порт TCP или UDP не задан для протокола прикладного уровня, используемого в соединении. Возлагая ответственность за выбор на сервер, ALPN облегчает сценарии, где выбор сертификатов или перемаршрутизация соединения может зависеть от согласованного протокола.

Выбор протокола в открытом виде как часть согласования позволяет избежать в ALPN ложной уверенности в части возможности скрыть согласованный протокол до организации соединения. Если сокрытие протокола требуется, предпочтительным методом является повторное согласование после организации соединения, что обеспечивает истинные гарантии защиты TLS.

5. Вопросы безопасности

Расширение ALPN не влияет на безопасность организации сессий TLS и обмена данными приложений. ALPN служит для предоставления видимого извне маркера протокола прикладного уровня, связанного с соединением TLS. Исторически сложилось так, что связанный с соединением протокол прикладного уровня можно определить по номеру используемого порта TCP или UDP.

Разработчикам и редакторам документов, планирующим расширить реестр идентификаторов протоколов, следует учитывать, что в TLS версий 1.2 и ниже клиент передаёт эти идентификаторы в открытом виде. Также следует учитывать, что ещё по меньшей мере 10 лет предполагается использование браузерами ранних версий TLS в начальных сообщениях ClientHello.

Следует проявлять осторожность в случаях, когда такие идентификаторы могут раскрывать персональные данные или такая утечка может приводить к профилированию или утечке конфиденциальных сведений. Если что-либо из отмеченного относится к новому идентификатору протокола, такой идентификатор **не следует** применять в конфигурациях TLS, где он будет виден в открытой форме, в документы со спецификацией таких идентификаторов протоколов следует включать рекомендации по предотвращению небезопасного использования.

6. Взаимодействие с IANA

Агентство IANA добавило в реестр ExtensionType Values приведённую ниже запись.

```
16 application_layer_protocol_negotiation
```

Этот документ организует реестр для идентификаторов протоколов Application-Layer Protocol Negotiation (ALPN) Protocol IDs в рамках имеющегося раздела Transport Layer Security (TLS) Extensions. Записи этого реестра должны включать указанные ниже поля.

- Protocol: имя протокола.
- Identification Sequence: набор значений октетов, точно указывающих протокол (это может быть имя протокола в кодировке UTF-8 [RFC3629]).
- Reference: ссылка на документ со спецификацией протокола.

Записи добавляются в реестр по процедуре Expert Review [RFC5226]. Назначенному эксперту рекомендуется поощрять включение ссылки на постоянную и легкодоступную спецификацию, которая позволяет создавать функционально совместимые реализации соответствующего протокола. Исходные записи реестра приведены ниже.

```
Protocol:  HTTP/1.1
Identification Sequence:
   0x68 0x74 0x74 0x70 0x2f 0x31 0x2e 0x31 ("http/1.1")
Reference:  [RFC7230]

Protocol:  SPDY/1
Identification Sequence:
   0x73 0x70 0x64 0x79 0x2f 0x31 ("spdy/1")
Reference:
   http://dev.chromium.org/spdy/spdy-protocol/spdy-protocol-draft1

Protocol:  SPDY/2
Identification Sequence:
   0x73 0x70 0x64 0x79 0x2f 0x32 ("spdy/2")
Reference:
   http://dev.chromium.org/spdy/spdy-protocol/spdy-protocol-draft2
```

```
Protocol:  SPDY/3
Identification Sequence:
   0x73 0x70 0x64 0x79 0x2f 0x33 ("spdy/3")
Reference:
   http://dev.chromium.org/spdy/spdy-protocol/spdy-protocol-draft3
```

7. Благодарности

Особую пользу этому документу принесло расширение для согласования следующего протокола (Next Protocol Negotiation или NPN), описанное Adam Langley, а также обсуждения с Tom Wesselman и Cullen Jennings (Cisco).

8. Литература

8.1. Нормативные документы

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC7230] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, June 2014.

8.2. Дополнительная литература

[HTTP2] Belshe, M., Peon, R., and M. Thomson, "Hypertext Transfer Protocol version 2", Work in Progress[1], June 2014.

[RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, January 2008.

Адреса авторов

Stephan Friedl
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA
Phone: (720)562-6785
EMail: sfriedl@cisco.com

Andrei Popov
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
USA
EMail: andreipo@microsoft.com

Adam Langley
Google Inc.
USA
EMail: agl@google.com

Emile Stephan
Orange
2 avenue Pierre Marzin
Lannion F-22307
France
EMail: emile.stephan@orange.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

[1]Опубликовано в RFC 9113. *Прим. перев.*