Network Working Group
Request for Comments: 2890
Category: Standards Track

G. Dommety
Cisco Systems

September 2000

Расширения GRE для полей Key и Sequence Number

Key and Sequence Number Extensions to GRE

Статус документа

Этот документ содержит проект стандартного протокола для сообщества Internet и служит приглашением к дискуссии в целях совершенствования и развития протокола. Текущее состояние стандартизации и статус протокола можно посмотреть в документе Internet Official Protocol Standards (STD 1). Распространение документа не ограничивается.

Авторские права

Тезисы

GRE[1] задаёт протокол для инкапсуляции любого протокола с использованием произвольного протокола сетевого уровня. Этот документ описывает расширение, с помощью которого два поля Key и Sequence Number могут передаваться в заголовке GRE [1].

1. Введение

Текущая спецификация GRE [1] задаёт протокол для инкапсуляции любого протокола в произвольный протокол сетевого уровня. Этот документ описывает расширение, с помощью которого два поля Key и Sequence Number могут передаваться в заголовке GRE [1]. Поле Key предназначено для идентификации отдельного потока трафика в туннеле. Поле Sequence Number служит для поддержки упорядочения пакетов в туннеле GRE.

1.1. Терминология

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с RFC 2119 [3].

Кроме того, для обозначения требований спецификации используется определённый ниже термин.

Silently discard – отбрасывание без уведомления

Реализация отбрасывает дейтаграмму без дальнейшей обработки и без индикации ошибки отправителю. Реализации **следует** поддерживать возможность записи ошибки в системный журнал с включением содержимого отброшенной дейтаграммы, **следует** также учитывать такие события в статистике.

2. Расширение заголовка GRE

Формат заголовка пакета GRE [1] показан ниже.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C|       Reserved0       | Ver |         Protocol Type         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Checksum (необязательно) |     Reserved1 (необязательно) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Предлагаемый заголовок GRE имеет обновленный формат.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C| |K|S| Reserved0       | Ver |         Protocol Type         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Checksum (необязательно) |     Reserved1 (необязательно) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Key (необязательно)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Sequence Number (необязательно)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Key Present (бит 2)

Если флаг Key Present установлен (1), это говорит о наличии поля Key в заголовке GRE. В остальных случаях поле Key не включается в заголовок GRE.

Sequence Number Present (бит 3)

Установленный (1) флаг Sequence Number говорит о наличии поля Sequence Number в заголовке. В противном случае поле Sequence Number не включается в заголовок GRE.

Позиции флагов Key Present и Sequence Present выбраны для обеспечения совместимости с RFC 1701 [2].

2.1. Поле Key (4 октета)

Поле Key содержит 4-октетное число, которое задаётся инкапсулятором. Фактический метод выбора значения Key выходит за рамки этого документа. Поле Key предназначено для идентификации отдельного потока трафика в туннеле. Например, для маршрутизации пакета может требоваться отсутствующая в инкапсулированных данных информация о

[1]Generic Routing Encapsulation - базовая инкапсуляция маршрутных данных.

контексте. Поле Key обеспечивает передачу контекста и определяет логические потоки трафика между инкапсулятором и декапсулятором. Пакеты, относящиеся к одному потоку трафика, инкапсулируются с одинаковым ключом Key и конечная точка туннеля на стороне декапсуляции определяет связь пакетов с потом по значению поля Key.

2.2. Порядковый номер (4 октета)

Четырёхбайтовое значение поля Sequence Number задаётся инкапсулятором при установленном в заголовке флаге Sequence Number Present. Поле Sequence Number **должно** использоваться получателем для восстановления порядка передачи пакетов от инкапсулятора к получателю. Поле Sequence предназначено для обеспечения негарантированной, но упорядоченной доставки пакетов. Если установлен флаг Key present (бит 2), порядковые номера относятся к потоку, указанному полем Key. Отметим, что пакеты без флага порядкового номера могут чередоваться с пакетами, где этот флаг установлен.

Порядковые номера принимают значения от 0 до 2^{32}-1. Первая дейтаграмма передаётся с номером 0. Порядковый номер определяется счётчиком по модулю 2^{32}. Получатель хранит значение порядкового номера последнего декапсулированного пакета. При организации туннеля GRE для этого номера следует устанавливать значение 2^{32}-1.

При получении декапсулятором пакета с нарушением порядка доставки такой пакет **следует** отбрасывать без уведомления. Пакет считается нарушающим порядок, если номер принятого пакета не превышает номер последнего декапсулированного пакета. Номер пакета считается не превышающим номер последнего декапсулированного пакета, если его значение попадает в интервал, включающий последний принятый номер и 2^{31}-1 предшествующих ему значения, включительно.

Если пакет принят без нарушения порядка доставки, он декапсулируется. Пакет считается соблюдающим порядок доставки, если его номер на 1 (по модулю 2^{32}) больше номера последнего декапсулированного пакета или пакет не имеет порядкового номера (флаг S не установлен).

Если пакет не является ни соблюдающим, ни нарушающим порядок доставки, это говорит о пропуске порядковых номеров. Получатель может буферизовать незначительное число пакетов в попытке восстановить их исходный порядок. В таких случаях пакеты помещаются в буфер по их порядковым номерам. Если пакет получен с соблюдением порядка и декапсулирован, получателю следует обратиться к началу буфера для проверки наличия в нем пакета, который ожидается следующим. Если такой пакет имеется в буфере, получатель должен декапсулировать его и посмотреть следующий пакет в буфере. При этом номер последнего декапсулированного пакета следует устанавливать в соответствии с номером последнего пакета, который был декапсулирован из буфера.

Ни при каких обстоятельствах пакет не должен находиться в буфере более OUTOFORDER_TIMER миллисекунд. Если ожидание затянулось, получатель **должен** незамедлительно пройти буфер в порядке сортировки, декапсулируя пакеты (и игнорируя пропуски порядковых номеров), пока в буфере не исчерпаются пакеты, для которых время ожидания уже достигло OUTOFORDER_TIMER. Номер последнего декапсулированного пакета следует изменить с учётом результатов декапсуляции.

Получатель может ограничить число пакетов в каждом буфере для потока (пакеты с одинаковым значением поля Key). При получении пакета, который вызовет превышение заданного значения MAX_PERFLOW_BUFFER, пакет из начала буфера незамедлительно декапсулируется независимо от его порядкового номера и номер последнего декапсулированного пакета соответственно меняется. После этого вновь прибывший пакет помещается в буфер.

Отметим, что порядковый номер служит для обнаружения потери пакетов и/или восстановления исходного их порядка (при наличии буферизации) при его нарушении в процессе доставки. Опция порядковых номеров должна применяться подобающим образом — в частности, имеет смысл отказаться от неё при работе с туннелирования, имеющими вышележащий уровень с упорядоченной доставкой или устойчивый к нарушению порядка. Если только часть протоколов, передаваемых в туннеле GRE, требует упорядоченной доставки, устанавливать флаг S в заголовке GRE следует лишь для соответствующих пакетов.

Восстановление нарушенного порядка доставки **может** выполняться декапсулятором для повышения производительности и обеспечения устойчивости к нарушению порядка доставки пакетов в сети. Небольшой буфер для восстановления порядка (MAX_PERFLOW_BUFFER) может помочь в плане повышения производительности, когда вышележащий уровень использует сжатие с учётом состояния или шифрование. Поскольку состояние сжатия или шифрования будет сбрасываться при потере пакетов, незначительные нарушения порядка можно скомпенсировать за счёт буферизации.

3. Вопросы безопасности

Этот документ описывает расширение, с помощью которого два необязательных поля Key и Sequence Number могут включаться в заголовок GRE [1]. При использовании поля Sequence number возможна вставка пакетов с произвольными номерами и организация DoS-атаки[1]. Для защиты от таких атак **должны** применяться протоколы IPsec [4], защищающие заголовок GRE и туннелируемые данные. Для защиты заголовка GRE **должен** применяться протокол ESP[2] [5] или AH[3] [6]. Протокол ESP обеспечивает защиту данных (payload) IP, которые включают заголовок GRE. При использовании AH обеспечивается аутентификация всего пакета, за исключением изменяемых полей. Отметим, что поле Key не участвует в какой-либо сортировке или защите (несмотря на его название).

4. Взаимодействие с IANA

Этот документ не требует выделения значений агентством IANA и не требует согласования с IANA.

[1]Denial of Service - отказ в обслуживании.
[2]Encapsulating Security Payload.
[3]Authentication Header.

5. Благодарности

Этот документ создан на основе исходных идей авторов RFC 1701. Kent Leung, Pete McCann, Mark Townsley, David Meyer, Yingchun Xu, Ajoy Singh и многие другие внесли свой вклад в обсуждение. Автор благодарен всем этим людям.

6. Литература

[1] Farinacci, D., Li, T., Hanks, S., Meyer, D. and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[2] Hanks, S., Li, T, Farinacci, D., and P. Traina, "Generic Routing Encapsulation", RFC 1701, October 1994.

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

[4] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[5] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[6] Kent, S., and R. Atkinson, " IP Authentication Header", RFC 2402, November 1998.

Адрес автора

Gopal Dommety

Cisco Systems, Inc.

170 West Tasman Drive

San Jose, CA 95134

EMail: gdommety@cisco.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

Полное заявление авторских прав

Подтверждение

Финансирование функций RFC Editor обеспечено Internet Society.