Network Working Group
Request for Comments: 5082
Obsoletes: 3682
Category: Standards Track

V. Gill
J. Heasley
D. Meyer
P. Savola, Ed.
C. Pignataro

October 2007

Обобщенный механизм защиты на базе TTL (GTSM)

The Generalized TTL Security Mechanism (GTSM)

Статус документа

В этом документе содержится спецификация протокола, предложенного сообществу Internet. Документ служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации протокола вы можете узнать из документа Internet Official Protocol Standards (STD 1). Документ может распространяться без ограничений.

Аннотация

Использование значений времени жизни TTL[1] (IPv4) или счётчика интервалов Hop Limit (IPv6) для проверки того, что пакет происходит от смежного узла на соединительном канале используется многими новыми протоколами. В этом документе обобщается данный метод. Документ заменяет собой экспериментальный RFC 3682.

Оглавление

1. Введение....1
2. Базовые допущения GTSM....2
2.1. Согласование GTSM....2
2.2. Оценка изощрённости атак....2
3. Процедуры GTSM....3
4. Благодарности....3
5. Вопросы безопасности....3
5.1. Обманные TTL (Hop Limit)....3
5.2. Туннелированные пакеты....3
5.2.1. Туннелирование IP по протоколу IP....4
5.2.2. Туннелирование IP через MPLS....4
5.3. Атаки из канала....5
5.4. Вопросы, связанные с фрагментированием....5
5.5. Протокольные сессии с промежуточной пересылкой (Multi-Hop)....5
6. Заявление о применимости....6
6.1. Совместимость с ранними версиями....6
7. Литература....6
7.1. Нормативные документы....6
7.2. Дополнительная литература....6
Приложение A. Multi-Hop GTSM....6
Приложение B. Отличия от RFC 3682....7

1. Введение

Обобщенный механизм защиты на базе TTL (GTSM[2]) разработан для защиты базирующейся на протоколе IP инфраструктуры управления маршрутизаторами от атак, основанных на перегрузке CPU. Хотя использование криптографических методов может защитить инфраструктуру маршрутизации (например, BGP [RFC4271], [RFC4272]) от широкого класса атак, многие атаки, нацеленные на перегрузку CPU, можно предотвратить с помощью простого механизма, описываемого в этом документе. Отметим, что такой же метод используется для защиты от других атак, направленных на истощение ресурсов, включающих процессоры маршрутизаторов, таких как атаки с перегрузкой шины подключения процессорной платы.

Работа GTSM основана на том, что в большинстве протоколов партнерские отношения организуются между смежными маршрутизаторами. Т. е. в большинстве случаев партнёры соединены напрямую между их интерфейсами или, в худшем случае, используют для соединения петлевые интерфейсы (loopback) со статическими маршрутами. Поскольку подмена[3] TTL считается почти невозможной, механизм основанный на ожидаемом значении TTL, может обеспечивать простую и достаточно отказоустойчивую защиту от атак на инфраструктуру, основанных на использовании обманных пакетов, передаваемых извне. Отметим, однако, что GTSM не является заменой механизмов аутентификации. В частности, этот метод не обеспечивает защиты от внутренних атак, связанных с доступом к каналам (например, с использованием обманных пакетов или повторным использованием собранных пакетов).

[1]Time to Live.
[2]Generalized TTL Security Mechanism.
[3]В оригинале - spoofing. *Прим. перев.*

Механизм GTSM одинаково применим к TTL (IPv4) и Hop Limit (IPv6). Более того, с точки зрения GTSM семантика TTL и Hop Limit идентична. Поэтому в оставшейся части документа термин «TTL» используется для обозначения как TTL, так и Hop Limit.

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

2. Базовые допущения GTSM

Работа GTSM основана на нескольких допущениях, перечисленных ниже.

1. Большая часть партнерских отношений организуется между соседними (смежными) маршрутизаторами.
2. Сервис-провайдеры могут использовать или не использовать строгую входную фильтрацию [RFC3704] на недоверенных каналах. Если требуется максимальная защита, такая фильтрация нужна (см. параграф 2.2).
3. Использование GTSM является **необязательным** и может настраивается на уровне отдельных пар партнёров.
4. Оба маршрутизатора-партнера поддерживают GTSM.
5. Маршрутизатор поддерживает метод разделения ресурсов (очередей, квот на обработку) для разных типов трафика.

Отметим, что в этом документе не описываются дополнительные ограничения, которые маршрутизаторы могут применять к пакетам, не соответствующим правилам фильтрации GTSM, типа отбрасывания пакетов, не соответствующих ни одной из заданных в конфигурации сессий, и ограничение скорости для остальных пакетов. Этот документ также не предполагает способов разделения ресурсов, поскольку такие способы зависят от оборудования и реализации.

Однако возможность предотвращения DoS-атак[1] основана на допущении о классификации пакетов и разделении их путей до того, как пакет начнёт использовать дефицитный ресурс системы. На практике чёткая GTSM со скоростью среды передачи обеспечивает наибольшую устойчивость систем к DoS-атакам.

2.1. Согласование GTSM

В этом документе предполагается, что при использовании существующих протоколов GTSM будет вручную настраиваться для каждой пары протокольных партнёров. Т. е., не предполагается и не определяется способов автоматического согласования GTSM, типа определённых в RFC 3392 [RFC3392].

Если новый протокол разрабатывается со встроенной поддержкой GTSM, рекомендуется всегда использовать этот механизм для передачи пакетов и проверки полученных пакетов (механизм GTSM всегда включён, см., например, [RFC2461]). Однако, если требуется динамическое согласование GTSM, протокольные сообщения, используемые для такого согласования, **должны** аутентифицироваться с использованием других механизмов защиты для предотвращения DoS-атак.

Отметим также, что в данной спецификации не предлагается базового механизма согласования возможностей GTSM, поэтому необходимо использовать протокольные сообщения с добавлением GTSM, если динамическое согласование представляется необходимым.

2.2. Оценка изощрённости атак

В этом документе предполагается, что атакующий достаточно изощрён и имеет доступ к точке, откуда он может передавать трафик управления в протокольный сеанс и этот трафик похож на нормальный трафик управления (т. е. пакеты имеют корректные для данной сессии адреса отправителя и получателя).

Предполагается также, что каждый маршрутизатор на пути между атакующим и объектом атаки корректно уменьшает значение TTL (естественно, что при компрометации пути или смежного партнёра, ситуация ухудшается).

Для обеспечения максимальной защиты следует использовать фильтрацию пакетов на входе до того, как пакет начнёт использовать дефицитный ресурс. В противном случае атакующий, подключенный непосредственно к интерфейсу, сможет нарушить работу защищённой с помощью GTSM сессии на этом или другом интерфейсе. Интерфейсы, для которых такая фильтрация не настроена (например, магистральные каналы) предполагаются недоступными для таких атак (т. е. расположенными в доверенной среде).

В качестве конкретного примера такого интерфейса мы полагаем, что туннель не имеет «чёрного хода», который позволил внедрять пакеты протокола с подставным TTL в защищённую GTSM сессию с непосредственно подключённым соседом. Предполагается, что 1) нет туннелируемых пакетов, адресованных маршрутизатору, 2) туннели, завершающиеся на маршрутизаторе, считаются защищёнными, а конечные точки - доверенными, 3) декапсуляция туннеля включает предотвращение подмены адреса отправителя [RFC3704], 4) поддерживающие GTSM сессии не позволяют протокольным пакетам приходить из туннеля.

Поскольку основные преимущества партнёрства реализуются между смежными маршрутизаторами, мы можем установить для пакетов протокола значение TTL = 255 (максимальное значение для IP) и отбрасывать пакеты протокола, которые приходят от партнёра со значением TTL **не равным** 255.

GTSM можно отключить для отдельных приложений типа маршрутных серверов или иных случаев партнёрства через несколько интервалов маршрутизации. В этом случае атака, ведущаяся через скомпрометированное соединение, может полностью нарушить партнёрство через это соединение.

[1] Denial-of-service - отказ в обслуживании.

3. Процедуры GTSM

Если механизм GTSM не встроен в протокол и используется в качестве дополнения (например, для BGP, LDP или MSDP), его **не следует** включать по умолчанию для того, чтобы обеспечивалась совместимость с не изменёнными вариантами протоколов. Однако, если протокол определяет встроенное динамическое согласование для GTSM, партнёр **может** предложить использование GTSM, которое будет включено при согласии обоих партнёров.

Если механизм GTSM включён для протокольного сеанса, к процедурам передачи и приёма пакетов IP добавляются перечисленные ниже действия.

При передаче пакетов.

- Поле TTL во всех пакетах IP, используемых для передачи сообщений, связанных с защищённой GTSM сессией протокола, **должно** иметь значение 255. Это требование относится также к сообщениям ICMP, связанным с обработкой ошибок.

- В некоторых вариантах архитектуры значение TTL для пакетов управления декрементируется модулем пересылки. Для сеансов со включённой защитой GTSM значения TTL **недопустимо** уменьшать.

При получении пакетов.

- Этап идентификации пакетов GTSM связывает каждый принятый пакет, который адресован уровню управления, с одной из трёх категорий доверия:

  - Unknown (неизвестный) - пакеты, которые невозможно связать ни с одной зарегистрированной сессией, поддерживающей GTSM, и, следовательно, GTSM не может определить уровень риска, связанного с таким пакетом;

  - Trusted (доверенный) - пакет идентифицирован, как относящийся к одной из поддерживающих GTSM сессий, и значение TTL лежит в допустимом диапазоне;

  - Dangerous (опасный) - пакет идентифицирован, как относящийся к одной из поддерживающих GTSM сессий, но значение TTL **не** относится к допустимому диапазону и, следовательно, GTSM предполагает наличие риска того, что пакет может быть обманным.

- Точные правила, применимые к пакетам разных категорий, не постулируются в этом документе и предполагаются настраиваемыми. Возможность настройки очевидно требуется (в частности) для сопутствующих пакетов (сообщений ICMP об ошибках). Следует отметить, что фрагментация может ограничивать объем информации, которая может быть доступна для классификации пакетов.

- Однако, по умолчанию каждой реализации:

  - **следует** обеспечить отсутствие конкуренции за доступ к ресурсам между пакетами категории Dangerous и пакетами категории Trusted или Unknown;

  - **недопустимо** отбрасывать (в процессе обработки GTSM) пакеты категории Trusted или Unknown;

  - **можно** отбрасывать пакеты категории Dangerous.

4. Благодарности

Использование TTL для защиты BGP рассматривалось множеством авторов, включая Paul Traina и Jon Stewart. Ryan McDowell предложил похожую идею. Steve Bellovin, Jay Borkenhagen, Randy Bush, Alfred Hoenes, Vern Paxon, Robert Raszuk и Alex Zinin предоставили полезные отклики на ранние версии этого документа. David Ward представил обобщение исходной идеи, связанной с BGP. Alex Zinin, Alia Atlas и John Scudder предоставили множество откликов для новой версии документа. Во время процедуры IETF Last Call и после неё были получены полезные комментарии от Francis Dupont, Sam Hartman, Lars Eggert и Ross Callon.

5. Вопросы безопасности

GTSM представляет собой простую процедуру, которая защищает протокольные сессии, организованные на одном интервале пересылки (single-hop), за исключением ситуаций, когда партнёр скомпрометрирован. В частности, этот метод не обеспечивает защиты против широкого класса атак on-the-wire (с прямым подключением к линии), для которых требуются более изощрённые механизмы.

5.1. Обманные TTL (Hop Limit)

Описанный здесь подход основан на наблюдении, что значение TTL (или Hop Limit) 255 не так просто подделать, поскольку по мере прохождения пакета по пути к адресату каждый маршрутизатор будет уменьшать значение TTL. В результате, при получении пакета маршрутизатором пригодность пакета IP проверить нельзя, но можно определить число маршрутизаторов, через которые он прошёл (в предположении, что ни один из маршрутизаторов на этом пути не был скомпрометирован для подмены значений TTL).

Отметим однако, что хотя создание пакетов с определенным значением TTL (по прибытии), происходящих из произвольной точки, сложно (но возможно), создать пакеты с TTL 255 при отсутствии непосредственного соединения невозможно (опять-таки в предположении отсутствия скомпрометированных соседей с непосредственным соединением и туннелей до декапсулятора, а также работы промежуточных маршрутизаторов в соответствии с RFC 791 [RFC0791]).

5.2. Туннелированные пакеты

Защита при любом методе туннелирования зависит от сложности аутентификации на конечных точках туннеля, а также от способа защиты туннелируемых пакетов «в полете». Однако эти механизмы выходят за рамки данного документа.

Сложность подделки пакетов с TTL 255 можно преодолеть, если для этих пакетов и самого туннеля не обеспечивается защиты целостности (т. е. нижележащий уровень скомпрометирован).

Когда пакет туннелируется непосредственно к протокольному партнёру (т. е. этот партнёр является декапсулятором туннеля), GTSM обеспечивает ограниченную защиту, которая зависит от целостности туннеля.

Если протокольная смежность организована через туннель и сам туннель представляется защищённым (т. е. нижележащая инфраструктура представляется защищённой, а туннель обеспечивает защиту от подделок для ключей и криптографических средств), GTSM может применяться для обнаружения протокольных пакетов, отправленных из точки, не являющейся другим концом туннеля. В дополнение к этому GTSM может помочь предотвратить атаки из-за соседнего маршрутизатора, если протокольный партнёр получает пакеты для защищённой с помощью GTSM сессии извне туннеля.

Когда конечная точка туннеля декапсулирует протокольный пакет и потом пересылает пакет IP протокольному партнёру, значение TTL уменьшается как описано выше. Это означает, что декапсулятор туннеля с точки зрения защищённого с помощью GTSM протокольного партнёра является предпоследним узлом. В результате проверка GTSM защищает от атакующих, которые инкапсулируют пакеты для ваших партнёров. Однако имеются особые случаи , когда соединение между декапсулятором туннеля и протокольным партнёром не включает интервал пересылки (hop) IP, где значение TTL уменьшается (например, туннелирование на канальном уровне, мост и т. п.). В архитектуре IPsec [RFC4301] ещё одним примером служит использование устройств BITW[1] [BITW].

5.2.1. Туннелирование IP по протоколу IP

Пакеты протокола могут туннелироваться через IP непосредственно протокольному партнёру или декапсулятору (конечная точка туннеля), который пересылает пакеты подключённому к нему непосредственно протокольному партнёру. Примерами туннелирования IP по протоколу IP являются IP-in-IP [RFC2003], GRE [RFC2784] и разные формы IPv6-in-IPv4 (например, [RFC4213]). Здесь возможны два варианта, которые проиллюстрированы ниже.

```
Партнёр ------------ Маршрутизатор конечной точки туннеля и партнёр
TTL=255      [туннель]   [TTL=255 на входе]
                         [TTL=255 при обработке]

Партнёр ------------ Маршрутизатор конечной точки туннеля ----- Партнёр на канале
TTL=255     [туннель] [TTL=255 на входе]                          [TTL=254 на входе]
                      [TTL=254 на выходе]
```

В обоих случаях инкапсулятор (исходная точка туннеля) является (предполагаемым) отправителем. Значение TTL во вложенной дейтаграмме IP может быть установлено в 255, поскольку RFC 2003 задаёт приведённое ниже поведение.

> При инкапсуляции дейтаграммы значение TTL во внутреннем заголовке IP уменьшается на 1, если туннелирование осуществляется, как часть пересылки дейтаграммы. В противном случае значение TTL во внутреннем заголовке при инкапсуляции не меняется.

В первом случае инкапсулируемых пакет туннелируется напрямую протокольному партнёру (конечная точка туннеля) и, следовательно, значение TTL в принятых протокольным партнёром пакетах может быть любым, включая 255.

Во втором случае инкапсулированный пакет туннелируется декапсулятору (конечная точка туннеля), который потом пересылает пакет непосредственно подключённому к нему протокольному партнёру. Для туннелей IP-in-IP документ RFC 2003 задаёт описанное ниже поведение декапсулятора.

> Значение TTL во внутреннем заголовке IP не меняется при декапсуляции. Если после декапсуляции во внутреннем заголовке TTL = 0, декапсулятор **должен** отбросить дейтаграмму. Если после декапсуляции дейтаграмма пересылается через один из сетевых интерфейсов декапсулятора, значение TTL будет уменьшаться в результате обычной пересылки IP. Дополнительная информация о работе с полем TTL приведена в параграфе 4.4.

Аналогично для туннелей GRE документ RFC 2784 задаёт указанное ниже поведение.

> Когда конечная точка туннеля декапсулирует пакет GRE, в который вложен пакет IPv4, адрес получателя из заголовка пакета IPv4 **должен** использоваться для дальнейшей пересылки пакета, а значение TTL в заголовке вложенного пакета **должно** быть декрементировано.

Следовательно, значение TTL в заголовке внутреннего пакета IP, видимое декапсулятору, может быть произвольным (в частности, 255). Если декапсулятор является и протокольным партнёром, ему можно будет доставить пакет с TTL 255 (первый случая). Если же декапсулятор должен переслать протокольный пакет непосредственно подключённому к нему партнёру, значение TTL будет уменьшено (второй случай).

5.2.2. Туннелирование IP через MPLS

Протокольные пакеты могут также туннелироваться партнёрам через MPLS LSP[2] как показано ниже.

```
Партнёр    --------   Завершающий LSP маршрутизатор и партнёр
TTL=255    MPLS LSP   [TTL=x на входе]
```

MPLS LSP может работать в режиме туннелирования Uniform или Pipe. Обработка TTL для этих режимов описана RFC 3443 [RFC3443], который обновляет RFC 3032 [RFC3032] в части обработки TTL в сетях MPLS. В RFC 3443 описана обработка TTL в режимах Uniform и Pipe, которые, в свою очередь, могут использовать или не использовать PHP[3]. Обработка TTL в этих случаях даёт разные результаты, поэтому они анализируются раздельно в параграфах 3.1- 3.3 RFC 3443.

Основное различие в плане обработки TTL между режимами Uniform и Pipe на завершающем LSP узле состоит в способе определения входящего значения TTL (iTTL). Для LSP в режиме Uniform iTTL будет принимать значение поля

[1]Bump-in-the-Wire - букв., «утолщение в проводе» - шифратор на канальном или физическом уровне, включаемый просто в разрыв сетевого кабеля.
[2]Label Switched Path - путь с коммутацией по меткам.
[3]Penultimate hop popping - выталкивание на предпоследнем интервале.

TTL из заголовка инкапсуляции (popped MPLS header), а для LSP в режиме Pipe iTTL будет принимать значение поля TTL из инкапсулированного заголовка.

Для Uniform LSP в RFC 3443 сказано, что на входе:

> Для каждой вытолкнутой метки в режиме Uniform значение TTL копируется из расположенного непосредственно ниже пакета IP/метки.

С этого момента внутреннее значение TTL (т. е. TTL в туннелируемой дейтаграмме IP) не содержит осмысленной информации и на краевом узле или в процессе PHP входное значение TTL (iTTL) будет равно TTL из вытолкнутого заголовка MPLS (параграф 3.1 в RFC 3443). Следовательно для Uniform LSP с несколькими (более 1) интервалами пересылки TTL на входе (iTTL) будет меньше 255 (x <= 254) и описанная в разделе 3 проверка даст отрицательный результат.

Трактовка TTL идентична для Short Pipe LSP без PHP и Pipe LSP (только без PHP). Для этих случаев в RFC 3443 сказано:

> «Для каждой вталкиваемой (pushed) метки в режиме Pipe или Short Pipe в поле TTL устанавливается значение, заданное оператором. Во многих реализациях по умолчанию установлено значение 255.»

В этих моделях трактовка пересылки на выходе основана на туннелированных, а не инкапсулированных пакетах. Входное значение TTL (iTTL) является значением поля TTL из видимого (exposed) заголовка, т. е. TTL туннелируемой дейтаграммы TTL. Следовательно, значение TTL в протокольных пакетах, видимое в точке завершения LSP, может быть произвольным (включая 255). Если завершающий LSP маршрутизатор является и протокольным партнёром, протокольные пакеты могут доставляться с TTL 255 (x = 255).

Для Short Pipe LSP с PHP значение TTL в туннелируемых пакетах не меняется после операции PHP. Поэтому в данном случае применимы те же выводы, которые были сделаны для Short Pipe LSP без PHP и Pipe Model LSP (только без PHP). Для Short Pipe LSP значение TTL на выходе не зависит от применения PHP.

В заключение отметим, что проверка GTSM возможна для пакетов IP, туннелируемых через Pipe LSP, но не через Uniform LSP. Кроме того, доставка протокольному партнёру пакетов протокола с TTL 255 невозможна, если завершающему LSP маршрутизатору требуется пересылать пакеты непосредственно подключённому к нему протокольному партнёру. Если пакет пересылает, выходное значение TTL (oTTL) определяется путём уменьшения iTTL на 1.

5.3. Атаки из канала

Как было описано в разделе 2, атакующий, подключившийся напрямую к одному из интерфейсов, может нарушить работу защищённой с помощью GTSM сессии на том же или другом интерфейсе (подменив адрес партнёра GTSM), если на интерфейсе не используется входной фильтрации. В результате интерфейсам без такой защиты приходится просто доверять в части отсутствия атак с их стороны.

5.4. Вопросы, связанные с фрагментированием

Как уже было отмечено, фрагментирование может ограничить объем данных, доступных для классификации. Поскольку фрагменты IP (кроме первых) не содержат информации уровня 4, очевидна невозможность связать их с зарегистрированной сессией GTSM. В соответствии с процедурами принимающего протокола, описанными в разделе 3, фрагменты IP, не являющиеся начальными, будут вероятней всего отнесены к типу Unknown. А поскольку для обработки пакета IP требуется собрать его из фрагментов, конечным результатом в сессии GTSM будет трактовка собранного пакета, как Unknown.

В принципе, реализация может запомнить значения TTL всех полученных фрагментов. После этого при сборке пакета проверяется соответствие TTL каждого фрагмента требуемому значению для связанной сессии с поддержкой GTSM. В очевидном общем случае когда реализация не проверяет все фрагменты, вполне возможно объединение легитимного первого фрагмента (который прошёл проверку GTSM) с поддельными последующими фрагментами с допущением того, что целостность принятого пакета не проверена и не защищена. Если проверка выполняется при сборке для всех фрагментов и неких фрагмент не прошёл проверку GTSM для сессии с поддержкой GTSM, собранный в результате пакет считается «опасным и недоверенным» (Dangerous-trustworthiness) с соответствующей этому обработкой.

Кроме того для сборки требуется дождаться получения всех фрагментов и это делает неприменимым допущение п. 5 в параграфе 2 - классификация не являющихся первыми фрагментов может оказаться невозможной по причине нехватки системных ресурсов, поскольку фрагменты потребуется буферизовать, а потом обработать с участием CPU. Т. е. в тех случаях, когда классификация не может быть выполнена с нужной детализацией, отличные от начальных фрагменты в сессиях с поддержкой GTSM не будут использовать разные пулы ресурсов.

Следовательно, для обеспечения на практике защиты от атак с применением фрагментов оператору может потребоваться ограничить скорость приёма или отбросить все полученные фрагменты. В таких случаях настоятельно **рекомендуется** для защищённых GTSM протоколов предотвращать фрагментацию и сборку путём ручной настройки MTU с использованием адаптивных измерений типа PMTUD[1] или иных доступных методов [RFC1191], [RFC1981] или [RFC4821].

5.5. Протокольные сессии с промежуточной пересылкой (Multi-Hop)

GTSM может обеспечить некоторую (трудно оцениваемую количественно) степень защиты при использовании в протокольных сессиях с промежуточной пересылкой (multi-hop, см. Приложение A). Чтобы избежать сложностей с количественной оценкой защиты и связанной с этим применимости метода здесь описан только вариант без промежуточной пересылки (single-hop), поскольку для него проще разобраться с защитой.

[1]Path MTU Discovery - определение MTU для пути.

6. Заявление о применимости

Механизм GTSM применим лишь для ограниченного числа топологий (и наиболее эффективен при прямом соединении протокольных партнёров.). В частности, применение метода следует ограничивать ситуациями, в которых протокольные партнёры соединены между собой напрямую.

GTSM не обеспечивает защиты от атакующих, которые расположены защищаемых узлов, как их легитимные партнёры. Например, если легитимные партнёры располагаются на расстоянии в один интервал пересылки (hop), GTSM не сможет обеспечить защиты от атакующих с непосредственно подключённых к тому же интерфейсу устройств (см. параграф 2.2).

Для соединений с промежуточной пересылкой (multi-hop) применимость GTSM требует дополнительных экспериментов и оценки защиты. Предполагается, что механизм GTSM будет пригоден для случаев наличия промежуточной пересылки, если топология соединения между партнёрами известна и не меняется, а промежуточные сети (между партнёрами) являются доверенными.

6.1. Совместимость с ранними версиями

RFC 3682 [RFC3682] не задаёт способов обработки «связанных сообщений» (ошибки ICMP). Данная спецификация задаёт установку и проверку TTL=255 для таких пакетов, как и для пакетов основного протокола.

Установка TTL=255 в пакетах связанных сообщение не создаёт проблем для реализаций RFC 3682.

Требование установки TTL=255 в пакетах связанных сообщений может оказывать влияние на реализации RFC 3682 в зависимости от используемого такой реализацией по умолчанию значения TTL (в некоторых принято 255, а в других - 64). Связанные сообщения второй категории реализаций RFC 3682 (не 255) будут считаться опасными (Dangerous) и обрабатываться в соответствии с разделом 3. Это не создаёт существенных проблем, поскольку протоколы не зависят от связанных сообщений (например, для разрыва сессии применяется протокольный обмен, а не TCP-RST) и доставка связанных сообщений не считается надёжной. Связанные сообщения, как таковые, обычно служат для оптимизации и сокращения тайм-аутов keepalive. Тем не менее, вспомогательные сообщения обеспечивают важный вектор атак (например, позволяя сбрасывать сессии), предлагаемое ограничение представляет обоснованным.

7. Литература

7.1. Нормативные документы

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.

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

[RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[RFC3392] Chandra, R. and J. Scudder, "Capabilities Advertisement with BGP-4", RFC 3392, November 2002.

[RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, January 2003.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

7.2. Дополнительная литература

[BITW] "Thread: 'IP-in-IP, TTL decrementing when forwarding and BITW' on int-area list, Message-ID: <Pine.LNX.4.64.0606020830220.12705@netcore.fi>", June 2006, <http://www1.ietf.org/mail-archive/web/int-area/current/msg00267.html>.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

[RFC3682] Gill, V., Heasley, J., and D. Meyer, "The Generalized TTL Security Mechanism (GTSM)", RFC 3682, February 2004.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.

[RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, January 2006.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

Приложение A. Multi-Hop GTSM

Примечание. Это приложение не является нормативной частью спецификации.

Основным применением GTSM являются непосредственные соединения между партнёрами. GTSM можно использовать для сессий с промежуточными узлами пересылки, где получатели будут проверять отличие значения TTL от 255 на заданное число промежуточных устройств. Поскольку применимость такого варианта менее очевидна и не столь понятно его влияние на защиту, этот случай не включён в спецификацию.

Приложение B. Отличия от RFC 3682

- Статус поднят до уровня Standards Track (RFC 3682 имел статус Experimental).
- Добавлен текст о применимости GTSM и использовании новых и существующих протоколов.
- Ограничена область действия сессиями без промежуточных узлов (без multi-hop).
- Явно указано, что связанные сообщения (ошибки ICMP) также должны передаваться и проверяться на наличие TTL=255. Вопросы совместимости с прежней версией рассмотрены в параграфе 6.1.
- Приведены разъяснения в части фрагментирования, использования туннелей и влияния входных фильтров.
- Внесены многочисленные редакторские правки для улучшения и прояснения текста.

Адреса авторов

Vijay Gill
EMail: vijay@umbc.edu

John Heasley
EMail: heas@shrubbery.net

David Meyer
EMail: dmm@1-4-5.net

Pekka Savola (editor)
Espoo
Finland
EMail: psavola@funet.fi

Carlos Pignataro
EMail: cpignata@cisco.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru