

Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Общие сведения](#)

[Механизм работы](#)

[Очереди AAAMGR](#)

[Ограничения](#)

[Связанные обсуждения Сообщества Cisco Support](#)

Введение

Этот документ описывает Регулировку AAA функция записей (RADIUS), которая поддерживает регулировку доступа (проверка подлинности и авторизация) и учетные записи, которые передаются серверу RADIUS.

Когда существует недостаточная пропускная способность для размещения suddenburst записей, генерируемых от маршрутизатора Cisco до сервера RADIUS, эта функция позволяет пользователю настраивать соответствующую скорость регулировки для предотвращения перегрузки сети и нестабильности.

Предварительные условия

Требования

Для этого документа отсутствуют особые требования.

Используемые компоненты

Сведения в этом документе основываются на ASR5k platform.

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

Общие сведения

Когда aaamgr передает сообщения RADIUS к серверу RADIUS в высокой скорости (например, когда большое число сеансов выключается в то же время, бухгалтерские сообщения остановки для всех сеансов генерируются в то же время), сервер RADIUS может не быть в состоянии получить сообщения в таких высоких скоростях. Для обработки этого условия, нам нужен эффективный механизм регулирования скорости в aaamgr, так, чтобы aaamgr передал сообщения на оптимальной скорости, таким образом, что сервер RADIUS

способен к получению всех сообщений и гарантирует, что никакое сообщение не отброшено из-за по загрузке в сервере RADIUS.

Механизм работы

Когда aaatgr передает сообщения в настроенной скорости к серверу RADIUS, это передает сообщения равномерно повсюду

каждую секунду вместо того, чтобы передать все сообщения в одиночном пакете. В зависимости от конфигурации каждая секунда разделена на несколько равных временных интервалов (с определенным периодом времени на слот). Минимальный период времени слота

могли быть 50 миллисекунд.

Скорость должна быть настроена, приняв их во внимание

- Скорость входящих вызовов,
- Количество aaatgr экземпляров
- Скорость, на которой сервер RADIUS может получить сообщения и
- Интервал Промежуточных периодов (для бухгалтерской конфигурации)
- Алгоритм используется для выбора сервера

Если установленное значение для значения серверов проверки подлинности будет слишком низко, то будет приводящее горлышко бутылки

перегрузка, которая может привести к вызовам, отбрасываемым должный открыть сеанс таймаут настройки. Если низкое значение будет настроено для учетных серверов, то большое удаление учета сообщений будет наблюдаться, из-за переполнения очереди.

Когда функция настроена, слоты временного интервала за секунду и период времени секунды вычислены и сохранены на уровне радиуса. Когда сообщение готово быть переданным серверу RADIUS, оно проверено, достигла ли квота (количество сообщений для на этот раз слота). Если предел не достигнут, сообщение передается, если это, то сообщение помещено в очередь в очереди уровня сервера, чтобы быть переданным в будущих временных интервалах. Каждый сервер RADIUS считает подробные данные о количестве сообщений передаваемыми в слоте текущего времени и время, в которое истекает временной интервал. Когда сообщения с очередями выбраны от очереди уровня сервера, они помещены в головку очереди уровня экземпляра, гарантировав предпочтение старым сообщениям, чем какое-либо другое новое сообщение. Сообщения от очереди уровня экземпляра выбраны для обслуживания.

Очереди AAAMGR

Существует два типа очередей в AAAMGR для сообщений:

1. Очереди уровня экземпляра
2. Очереди уровня сервера

Когда сообщение генерируется, оно первоначально помещено в очередь в очереди уровня экземпляра для обслуживания.

Очередь уровня экземпляра обработана для 25 миллисекунд для каждых 50 миллисекунд.

Любое сообщение, которое исключено из очереди от очереди уровня экземпляра, будет предпринято, чтобы быть переданным серверу RADIUS. При некоторых условиях мы можем не быть в состоянии передать сообщения (никакая доступная пропускная способность или никакие доступные ID). В таких случаях сообщения, которые отказали попытку, будут помещены в очередь в очередях уровня сервера. Для каждые 50 миллисекунд вы выбираете как много сообщений, которые имеют в наличии ID и также доступную пропускную способность и помещают их во главе очереди уровня экземпляра (эти сообщения являются более старыми, чем какое-либо другое сообщение, которое присутствует в очереди уровня экземпляра).

Когда существует регулирование скорости для учета сообщений, и если существует большой учет сообщений в очереди уровня экземпляра, то любое новое сообщение аутентификации переходит к хвосту очереди уровня экземпляра. Для того, чтобы быть обработанным это должно ждать всего бухгалтерского сообщения (предшествующий новому сообщению аутентификации), чтобы быть или переданным серверу RADIUS или быть перемещенным к очереди уровня сервера. Это - существующее поведение, и это не модифицируется. Таким образом, это может заставить маленькую задержку нового сообщения аутентификации быть обработанной.

Пример

На основе Max. скорости со значением 5, можно передать пять сообщений через 1 секунду и иметь 256 выдающихся сообщений проверки подлинности RADIUS (Max. выдающаяся конфигурация по умолчанию) оставшийся без ответа на ааатгр к Серверу проверки подлинности RADIUS. В случае, если существует больше чем 5 сообщений за 1 секунду, сообщения помещены в очередь, пока AAA-сервер не отвечает на существующие запросы.

В случае, если вы достигаете 256 сообщений Проверки подлинности RADIUS, передаваемых от одного ааатгр к серверу, остающиеся запросы будут помещены в очередь, пока AAA-сервер не ответит на существующие запросы. Это снова войдет в ту же очередь как та из Max. скорости. Сообщение поднято с очереди только, когда у вас есть свободный слот. Свободный слот входит при получении ответа для сообщения или когда это испытывает таймаут.

Ограничения

Так как Cisco ASR5K является распределенной системой с независимыми sessmgr/ааатгр парами, обрабатывающими вызовы, регулировка скорости могла быть внедрена только для независимых ааатгр экземпляров. Это теоретически для расширения скорости единственного экземпляра ко всей коробке Cisco ASR5K в целом, просто умножив общее число экземпляров с Max. скоростью из каждого экземпляра.

Этот номер является просто абсолютным верхним пределом в солнечном дневном сценарии. Вы не можете рассматривать Cisco ASR5K как черный ящик и не можете предположить, что все вызовы должны успешно выполняться, если бы расчетное значение, замеченное в системе, не пересекло верхний предел.

Max. скорость радиуса связана другими внутренними и внешними параметрами, отнесенными к системе. Посмотрите ожидаемое влияние, если не соблюдают одно из условий.

Условия

Однородное распределение вызовов от demuxmgr до всего sessmgrs

Однородное распределение IMSI (это на всякий случай из циклического учета посредничества)

Никакие внезапные пакеты прибытия вызовов - в

Серверы RADIUS должны ответить вовремя

Влияние, если не встреченный с

Если распределение вызова не универсально, то сообщения RADIUS могут быть помещены в очередь для некоторых экземпляров. Таким образом даже при том, что теоретическое Max. ограничение скорости не достигнуто, вызовы будут отброшены для экземпляров, где сообщения помещены в очередь.

Считающий посредничество циклический алгоритм основывается на основанной на IMSI маршрутизации. В этом случае, на основе распределения IMSI, некоторый набор серверов может быть предпочтен по другим на основе логики маршрутизации, очередь могла бы быть составной для тех серверов, приводящих к сбросу вызовов. Если будет пакет новых вызовов, то с другой стороны недавно генерируемые сообщения RADIUS будут помещены в очередь в системе. К тому времени, когда новые запросы RADIUS обработаны. Подготовительное время сеанса может истечься, приведя к сбросам вызовов. Когда запросы RADIUS испытывают таймаут из-за неполадок сервера, там? И снова быть наращиванием очереди, потому что новые запросы не будут отосланы пока текущий, ожидая ответ не будет удален из системы. Скорость, на которой вызванные таймаут сообщения будут удалены из системы, зависит от Max. выдающегося и конфигураций таймаута также.

Во многих случаях мы видим, что запросы доступа не обработаны всеми активными aaamgr задачами. Это означает, что у нас есть неравномерное распределение вызова в sessmgr задачах и далее на, не, все aaamgr экземпляры включены в обработку вызовов.

Распределение вызова не на основе строгого кругового механизма, который является, если будет 10 входящих вызовов, то они перейдут к 10 sessmgrs в монотонном алгоритме.

Распределение вызова основывается на этих четырех основных факторах

- **active_session_count**
- **cpu_load**
- **Round_trip_delay** (demuxmgr? sessmgr? demuxmgr)
- **outstanding_add_request** (демультиплексируют к sessmgr),

Это - текущая реализация. Max. скорость является просто верхним пределом, но из-за распределенного характера нашей архитектуры, вы можете? t непосредственно экстраполируют его к загрузке шасси. Поведение зависит от загрузки на данном AAAmgr в установленный срок.

Max. последовательность оценок радиуса должна использоваться для **мониторинга статуса** системы. Если существует **наращивание очереди**, то это означает один из этих 4 (обратитесь к таблице), с условиями не соблюдают, и необходимо определить основную причину для того же.

** порог Max. последовательности оценок мог быть внедрен и постоянно проверяться.