НОВЫЙ ПОДХОД К ПОВЫШЕНИЮ БЕЗОПАСНОСТИ MPLS VPN ПУТЕМ ПРИНЯТИЯ ПРОГРАММНО-ОПРЕДЕЛЯЕМОЙ СЕТЕВОЙ ПАРАДИГМЫ

Рубинштейн Р. Ю.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Цветков В. Ю. – д. т. н., доцент

В работе рассмотрен новый подход к повышению безопасности MPLS VPN путем принятия программно-определяемой сетевой парадигмы

Безопасность сетевых инфраструктур является одной из утомительных задач в современных сетях. Действительно, в наши дни требуемая безопасность должна характеризовать динамизм и способность адаптироваться к контексту бирж, другими словами, безопасность не должна влиять на производительность сети. Чтобы удовлетворить эту потребность, можно использовать автоматизацию сети через контроллер программно-определяемой сети (SDN). SDN - новая парадигма, позволяющая через контроллер управлять всей архитектурой сети. В этой статье мы предлагаем новое решение для динамической генерации политик безопасности между различными сайтами MPLS VPN путем принятия подхода SDN. Безопасность является одной из основных задач бизнеса, поскольку безопасность - это не только конфиденциальность, целостность или аутентификация, но и высокая доступность. Высокая доступность зависит от нескольких факторов, а именно от используемого оборудования, стратегий и планов, предоставляемых компанией в случае неисправности системы. Иногда компании могут потребоваться четыре основных принципа безопасности, поэтому для поиска компромисса и решения, гарантирующего их, требуется много внимания. Например, протоколы шифрования или политики безопасности трафика в целом могут влиять на производительность оборудования и, таким образом, ставить под угрозу доступность ресурсов.

Многопротокольная коммутация по меткам «MPLS» рассматривается как основной протокол, развернутый на уровне ядра сети оператора. MPLS был успешным с появлением новых связанных сервисов, прежде всего сервиса виртуальной частной сети (VPN) . MPLS VPN позволяет получить безопасное соединение с меньшими затратами. Поэтому для создания клиентских VPN необходимо изолировать потоки каждого клиента.

Это правда, что MPLS VPN обеспечивает высокий уровень безопасности по сравнению с традиционными VPN, потому что трафик проходит через частную сеть оператора, но некоторые клиенты предпочитают добавлять уровень шифрования через протокол IPsec . IPsec также опирается на два протокола: 1) аббревиатуру заголовка аутентификации для AH, гарантирующую аутентификацию, целостность и защиту от повторного воспроизведения данных, 2) полезную нагрузку инкапсуляции "ESP", обеспечивающую большую конфиденциальность.

С появлением облака видим дополнительный шаг в автоматизации процессов с помощью Software Defined Network (SDN). Это значительно упрощает автоматизированные операции в стандартных и воспроизводимых средах. Благодаря этому новому режиму работы этапы тестирования и развертывания сокращены, что существенно экономит время и деньги. SDN позволяет по принципу оркестровки управлять сетевыми ресурсами компании из центральной точки, называемой контроллером. Парадигма SDN может быть принята для реализации новых правил для улучшения политик безопасности защищенных IPsec туннелей MPLS VPN с целью удовлетворения потребностей компании, особенно с точки зрения безопасности, целостности, аутентификации и особенно доступности.

Подход состоит из трех этапов: измерение производительности сети (приложения и оборудование), расчет соответствующей политики IPsec и развертывание этой политики на маршрутизаторах и устройствах. Эти шаги вращаются вокруг четырех элементов: доступность, конфиденциальность, целостность и аутентификация.

Список использованных источников:0

1. Bensalah, F., & El Kamoun, N. (2019). Novel software-defined network approach of flexible network adaptive for VPN MPLS traffic engineering. International Journal of Advanced Computer Science and Applications, 10(4), 280-284.

2. Bahnasse, A., Louhab, F. E., Ait Oulahyane, H., Talea, M., & Bakali, A. (2018). Novel SDN architecture for smart MPLS traffic engineering-DiffServ aware management. Future Generation Computer Systems, 87, 115-126.