УДК 623.618

А.В. Снегуров, В.Х. Чакрян

Харьковский национальный университет радиоэлектроники, Харьков

МЕХАНИЗМЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ СТЕКА ПРОТОКОЛОВ IPV6

В статье проводится анализ механизмов обеспечения безопасности семейства протоколов IPv6, таких как: SEND, IPv6 Firewall, IPv6 IPSec, SAVI, IPv6 Routing Advertisement Guard, IPv6 Source/Prefix Guard, IPv6 Destination Guard, IPv6 Snooping, DHCPv6 Guard. В статье проанализированы текущее состояние и нерешенные вопросы исследуемой области. Сделан акцент на то, что данные механизмы безопасности, в некоторых случаях, создают новые потенциальные векторы атак. Результаты статьи могут использоваться для разработки новых механизмов обеспечения информационной безопасности телекоммуникационных сетей.

Ключевые слова: *IPv6, механизмы безопасности IPv6, SEND, CGA, HBA, SAVI, IPv6 IPSec, IPv6 Firewall, IPv6 First Hop Security.*

Введение

Одной из задач проектирования межсетевого протокола IPv6 было повышение требований безопасности по сравнению со своим предшественником, протоколом IPv4. Так в основе протокола IPv6 лежит утверждение, что IPSec должен использоваться в данном протоколе по умолчанию. Однако, в большинстве практических случаев, при использовании IPv6, протокол IPSec не используется вообще. К тому же в новых протоколах, которые лежат в основе стека протоколов IPv6, найдено множество уязвимостей. С целью устранения таких уязвимостей были разработаны специальные механизмы безопасности:

- протокол CGA (англ. “Cryptographic generated address”) [1], который позволяет генерировать IPv6 адрес на основе криптографических алгоритмов и позволяет предотвратить подмену IPv6 адресов в сети;
- протокол SEND (англ. “Secure Neighbor Discovery protocol”) [2], который позволяет обезопасить стандартный NDP протокол путем применения криптографии.

Следует отметить, что первоначальные версии вышеприведенных механизмов безопасности в IPv6 сети были разработаны в 2005 году. В последние годы в связи с растущим спросом на новую версию IP протокола предлагаются усовершенствования и доработки данных механизмов. Основные принципы работы протокола SEND, а также CGA и их доработки более детально рассматриваются далее.

Помимо CGA и SEND разработаны и другие механизмы обеспечения безопасности, в частности:

- протокол SAVI (англ. “Source Address Validation Improvement”), который позволяет гарантировать валидность IP адресов в пределах локальной сети;
- группа механизмов обеспечения First-Hop безопасности: IPv6 RA Guard, IPv6 Snooping, DHCPv6 Guard, IPv6 Source Guard и Prefix Guard, IPv6 Destination Guard.

Целью статьи является анализ существующих механизмов обеспечения безопасности стека протоколов IPv6, а также проблем в данной области, нерешенных в настоящее время.

В данной статье не рассматриваются механизмы безопасности для Mobile IPv6, туннелирования IPv6-IPv4 трафика, множественной адресации (англ. “Multihoming”) IPv6, либо других функциональных возможностей IPv6, обеспечение безопасности которых выходит за рамки локальной сети.

Механизмы обеспечения безопасности

Каждый из механизмов безопасности требует более детального анализа, который позволит выявить оставшиеся уязвимые места в семействе протоколов IPv6 и найти возможные векторы атак в самом алгоритме защиты.

Протокол CGA

Данный механизм позволяет создать уникальный IPv6 адрес, который считается невозможным подменить за разумное время, не зная всех параметров, на основании которых генерируется CGA. Главная идея протокола CGA состоит в том, чтобы создать идентификатор интерфейса (правые 64 бита) IPv6 адреса при помощи вычисления криптографического хэша публичного ключа устройства. Затем приватный ключ может быть использован для подписи исходящих от устройства сообщений.

Для того чтобы проверить подпись получателю необходимо знать: исходный адрес, публичный ключ и значения дополнительных параметров. Данный процесс не связан с PKI инфраструктурой, не требует никаких дополнительных центров сертификации и доверительных сторон.

На рис. 1 отмечены несколько параметров, которые используются при создании CGA:

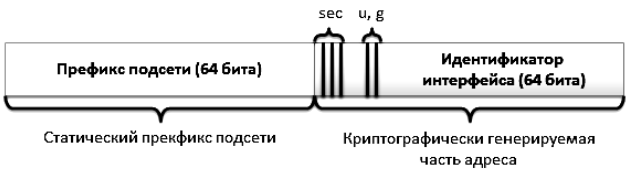

Рис. 1. Формат IPv6 адреса с учетом параметров, необходимых для генерации CGA

- Параметр sec занимает первые три наиболее значимых бита идентификатора интерфейса (англ. "Interface Identifier") и влияет на безопасность первого вычисляемого хэш-образа Hash1. При $sec = 0$ некоторые шаги алгоритма генерации CGA могут быть пропущены, что ускоряет процесс создания криптографического адреса, однако уменьшает его стойкость (табл. 1).
- Параметры $u = g = 1$ занимают место 7 и 8 наиболее значимых битов идентификатора интерфейса и используются для обозначения того, что данный IPv6 адрес создан на основе CGA.

Таблица 1

Зависимость стойкости CGA от параметра sec . Заимствовано из статьи [3]

значение sec	1	2	3	4	5	6
Время компрометации	0,2 с.	3,2 ч.	24 г.	$1{,}6\cdot10^{6}$ лет	10^{11} лет	$6{,}8\cdot10^{15}$ лет
Условия тестирования: для теста в качестве хэш генератора выбрана открытая OpenSSL библиотека и процессор AMD64. Скорость генерации $2^{18{,}5}$ хэшей / сек. на одном ядре процессора.						

В качестве функции хэширования в исходном стандарте [1] используется SHA-1, однако, прибегая к более детальному анализу CGA, некоторые авторы предлагают отойти от обозначения конкретной функции хэширования [3]. Также для генерации CGA необходимы следующие параметры:

- Модификатор [M] (англ. "Modifier") – случайное 128 битовое поле. Увеличивает приватность, внося случайность в процесс генерации адреса.
- Префикс подсети [ПП] (англ. "Subnet Prefix") - 64 битовое поле, обозначающее статический префикс подсети.
- Идентификатор интерфейса [ИИ] (англ. "Interface Identifier") - IPv6 часть адреса, которая идентифицирует конкретный интерфейс.
- Счетчик коллизий [СК] (англ. "Collision Count") – 8 битовое поле, которое может принимать значения 0, 1, или 2. Увеличивается в процессе создания адреса для восстановления после коллизий в случае обнаружения дублирующихся адресов.
- Публичный ключ [Пуб.кл.] (англ. "Public Key") – поле переменной длины, но не менее 384 битов, содержит публичный ключ владельца адреса (отправителя).
- Расширенные поля [Прив.кл.] (англ. "Extension Fields") – поля переменной длины, не используются на данный момент и могут быть использованы в будущих модификациях алгоритма. Формат полей описан в обновленном RFC 4581 [0].

Процедура генерации CGA состоит из следующих шагов:

1) Генерировать случайное 128 битовое значение M. Выбрать параметр sec и установить СК в ноль.

2) Конкатенировать M, $64+8$ нулевых битов, и Пуб.кл. Хэшировать полученное значение, 112 битов слева являются значением Hash2.

3) Сравнить n левых битов Hash2, где $n \in [0;16\cdot sec]$, на равенство нолю. Если они все равно нолю (либо если $sec = 0$), продолжаем алгоритм и выполняем шаг 4. В противном случае, увеличиваем M и возвращаемся к шагу 2.

4) Конкатенировать M, ПП, СК и Пуб.кл. Хэшировать полученное значение. Левые 64 бита хэша являются значением Hash1.

5) Создать корректный ИИ, выставив биты u, g в единицу и установив первые три бита такими же, как выбранный параметр sec.

6) Объединить ПП и ИИ для формирования полного IPv6 CGA адреса.

7) В случае если обнаружен такой же IPv6 адрес, то увеличить СК на 1, и вернутся к 4 шагу алгоритма. В случае если произошло более 3 коллизий - оповестить об ошибке и приостановить алгоритм.

Процедура проверки CGA состоит из следующих шагов:

1) Проверить, чтобы СК равнялся 0, 1 или 2, и что реальный ПП соответствует присланному ПП от узла. Проверка CGA является некорректной, в случае, если хотя бы одно условие не соблюдается.

2) Конкатенировать M, ПП, СК и Пуб.кл. Хэшировать полученное значение. Левые 64 бита хэша являются значением Hash1.

3) Сравнить Hash1 с ИИ адреса. Разница в битах u, g и трех крайних левых битов параметра sec игнорируются. Если 64-битовые значения отличаются (кроме как в позициях игнорируемых 5 битов), то проверка считается некорректной.

4) Объединить M, $64+8$ нулевых битов и Пуб.кл. Хэшировать полученное значение, 112 битов слева являются значением Hash2.

5) Считать параметр sec из трех крайних левых битов ИИ адреса. Сравнить $16\cdot sec$ наиболее левых битов значения Hash2 с нулем. Если хотя бы один из этих битов не равен нулю, проверка CGA считается некорректной. В противном случае проверка считается пройденной (если параметр sec ра-

вен нулю, то проверка всегда считается корректной на этом шаге).

В первоначальном алгоритме существовало несколько проблем и ограничений, к примеру, жесткое использование лишь SHA-1 в качестве функции хэширования. Также злоумышленник может генерировать точно такой же ИИ CGA адреса, как и настоящего узла, однако с другим префиксом подсети. Также с увеличением мощности компьютеров увеличивается шанс нахождения коллизий при вычислении хэшей, что приводит к возможному появлению одинаковых CGA в рамках одной подсети.

В статье [3] предлагается усовершенствовать алгоритм CGA и назвать его как CGA++. В улучшенном алгоритме предлагается создать подпись параметров, на основании которых генерируется CGA, а затем учесть эту подпись при формировании хэша для проверки целостности самой подписи. Предлагается осуществлять это следующими шагами:

- Подписать ключом Прив.кл. результат конкатенации М, СК и ПП. Результатом является цифровая подпись (ЦП) параметров CGA.
- Конкатенировать Пуб.кл. и ЦП. Произвести хэширование над результатом конкатенации. Наиболее значимые 64 бита являются значением Hash1.

Выше приведены только пункты, которые отличны в CGA++ по сравнению с CGA алгоритмом. Для просмотра полного алгоритма можно обратиться к оригинальной статье [3]. Также в данной статье проведен анализ безопасности и стойкости обоих алгоритмов. На данный момент CGA активно применяется в протоколе SEND [5], который обеспечивает безопасность NDP протокола. CGA в данном протоколе служит подтверждающим фактором тому, что отправитель NDP сообщения действительно является его владельцем. Также в [6] авторы предлагают использовать CGA в протоколе IKE2, что, по их мнению, упрощает схему взаимодействия между узлами.

Однако CGA не универсален. К примеру, для повышения безопасности протокола множественной адресации (англ. “multihoming”) предложено использовать хэш-генерированный адрес (англ. “Hash Based Address”, HBA). HBA значительно снижает потребление ресурсов, так как генерирует IPv6 адрес лишь с применением функций хэширования, в отличие от CGA, который основан на асимметричной криптографии [2].

HBA может работать автономно, либо в режиме совместимости с CGA. Их алгоритмы генерации адреса крайне похожи. Основное различие состоит в том, что в чистом HBA не генерируется пара открытого и приватного ключей. Более подробно о данном алгоритме можно прочесть в официальном стандарте, посвященном HBA [7].

Протокол SEND

Протокол SEND определяет механизмы обеспечения безопасности NDP протокола. В отличие от защитных механизмов, предлагаемых в оригинальном NDP протоколе, механизмы SEND не прибегают к использованию IPsec. Описание данных защитных механизмов SEND для обеспечения безопасности NDP протокола представлено в табл. 2.

Таблица 2

Механизмы защиты SEND против атак на NDP

Тип атаки	Защитные механизмы
Подделка NS/NA сообщений	SEND требует RSA подпись и использование CGA (включенная опция CGA) в NS сообщениях
Атака на процесс проверки недоступности соседнего узла	SEND требует от узла, который отвечает на NS сообщение включить механизм RSA подписи и доказательство авторизованного использования идентификатора интерфейса в адресе, с которым производится коммуникация
DoS-атака на процесс проверки дублирующегося адреса	SEND требует включить механизм RSA подписи и доказательство авторизации в NA сообщения, которые отправляются в ответ на DAD запрос
Подделка RA/RS сообщений	SEND требует включить механизм RSA подписи и доказательство авторизации RA сообщения
Атака повторного воспроизведения (англ. “Replay Attacks”)	SEND включает механизм Nonce в NS/RS сообщения и требует, чтобы в NA/RA сообщениях он также был активен. Для защиты от нежелательных NA/RA и Redirect сообщений используется механизм временной метки (англ. “Timestamp”).

В протоколе SEND описаны следующие механизмы безопасности:

- CGA

Механизм рассмотрен ранее в данной статье. Позволяет генерировать уникальный IPv6 адрес, который невозможно подделать в рамках одного префикса подсети.

- Механизм RSA подписи (англ. “RSA Signatures”)

Цифровая подпись открытого ключа позволяет обеспечить целостность сообщения и аутентифицировать отправителя. Подобная подпись должна прикрепляться к каждому NDP сообщению.

- Механизм временной метки

Временная метка позволяет защититься от атаки повторного воспроизведения.

Атака может быть воспроизведена в период, пока метка является активной (истекает через определенное время). Также атака на протокол синхронизации времени, к примеру, NTP (англ. “Network Time Protocol”), может изменить параметры времени на устройстве и открыть окно для атаки повторного воспроизведения.

- Механизм Nonce

Nonce – это псевдослучайное число, генерируемое хостом, которое используется лишь один раз. В

протоколе SEND параметр Nonce используется для проверки того, что определенный отклик действительно связан с запросом, который его породил.

- Запрос пути сертификации (англ. “Certification Path Solicitation, CPS”)

Хост должен знать как минимум один маршрутизатор, с которым у него установлен доверительный якорь (англ. “Trust Anchor”). Данный маршрутизатор является доверительной стороной, которая сертифицирует другие маршрутизаторы в сети. Сертификат указывает, какие префиксы подсети имеет право раздавать конкретный маршрутизатор. После выбора хостом определенного маршрутизатора в качестве шлюза и получения от него IPv6 адреса, хост совершает проверку пути сертификации (англ. “Certification Path”) данного маршрутизатора путем отправки запроса пути сертификации (англ. “Certification Path Solicitation, CPS”) и получения ответа пути сертификации (англ. “Certification Path Advertisement, CPA”). Проверка осуществляется таким же стандартным путем, как и в инфраструктуре с открытым ключом (англ. “Public Key Infrastructure, PKI”). Однако в данном случае роль центра сертификации (англ. “Certificate Authority, CA”) играет маршрутизатор, который является доверительным якорем для хоста.

Протокол SEND также подвержен атакам, в частности, в самом стандарте описаны атаки типа DoS, которые могут быть реализованы различными путями:

- отправить на устройство жертвы большое количество пакетов, которые заставят хост выполнить множество операций асимметричной криптографии, которые являются крайне трудоемкими для современных процессоров;
- устроить CGA атаку;
- также злоумышленник может, используя Authorization Delegation Discovery процесс, заставить маршрутизатор запросить большое количество путей сертификации к различным устройствам, которые играют роль якорей доверия. Методом борьбы с данной атакой может стать кеширование результатов прошлых запросов (в том числе негативных), тем самым, снижая потребность в генерации новых запросов;
- атакующий может отправить на хост множество путей сертификации, обработка которых потребует большое количество системных ресурсов.

На данный момент главным методом борьбы с данными атаками является учет, контроль и ограничение ресурсов, которые потребляются потенциально небезопасными процессами. Однако очевидно, что подобный подход является не идеальным. К примеру, ограничение потребления ресурсов для определенного процесса может привести к тому, что большое количество безопасных запросов будут отброшены. Также подобный метод оставляет возможность злоумышленнику совершить DoS-атаку на сам сервис, отправив большое количество запросов, тем самым, создавая препятствие для прохождения легитимного трафика.

IPv6 Firewall

Брандмауэр является передовым средством защиты любой современной сети. Данное устройство позволяет оградить вашу сеть от множества внешних угроз и создать базовый уровень защиты. С приходом IPv6 протокола появились новые угрозы, к которым должны адаптировать современные брандмауэры.

К примеру, IPv6 может использовать в одном пакете множество различных расширенных заголовков. Для того чтобы брандмауэр получил доступ к информации вышележащих уровней, ему необходимо обработать весь заголовок 3 уровня. В случае если злоумышленник специально отправляет пакеты с большим количеством вложенных расширенных заголовков, это может привести к чрезмерному потреблению ресурсов брандмауэра и, вследствие этого, к отказу в обслуживании.

В RFC 4942 [8] рассматриваются уязвимости, связанные, в том числе, с расширенными заголовками и фильтрацией пакетов, а также предлагаются решения для их предотвращения. Однако некоторые уязвимости с расширенными заголовками все же остались.

К примеру, в расширенном заголовке Hop-by-Hop можно создать неограниченное количество опций, тем самым, создавая большие пакеты, сложные для обработки [9]. Также в оригинальном стандарте нет ограничений на использование Pad1..PadN заполнителей для расширенных заголовков Hop-by-Hop и Destination option. Брандмауэр может бороться с подобными ситуациями, проверяя только часть расширенных заголовков до определенного уровня вложенности, однако в противном случае, в последующих заголовках злоумышленник может передавать нужные ему данные (пример, скрытые туннели, организованные с помощью расширенных заголовков).

Также брандмауэр должен контролировать фрагментированные IPv6 пакеты, так как в некоторых случаях фрагментация пакетов может использоваться для обхода брандмауэра и совершения атаки на сеть. Пример такой атаки описан в [10]. Она основана на том, что в пакет записывается расширенный заголовок Destination option, который перезаписывает информации для вышележащего протокола TCP. Брандмауэр должен корректно фильтровать поток сообщений и отбрасывать те, которые пытаются перезаписать информацию вышележащих уровней.

Также брандмауэр должен корректно фильтровать Tiny-Fragment – это такие IPv6 пакеты, в которых заголовок вышележащего уровня расположен во втором фрагменте пакета. Злоумышленник может использовать подобные запросы для обращения к портам, которые должны быть заблокированы брандмауэром.

В RFC 2460 [11] указано, что если все фрагменты пакета не были получены в течении 60 секунд, то пакет должен быть отброшен. Соответственно и

брандмауэр должен контролировать поток таким образом, чтобы фильтровать и не пропускать фрагменты пакета, которые передаются после 60 секунды.

Важные изменения в фильтрации коснулись и ICMP протокола. Существует ряд сообщений, которые невозможно блокировать в виду особенностей работы протокола IPv6. Например, сообщения, которые относятся к протоколу NDP, без которого узел не сможет получить IPv6 адрес, проверить его на уникальность и обнаружить соседей. Так как ICMP протокол является одной из наиболее важных составных частей IPv6, то в 2007 году был выпущен RFC 4890 [12], в котором описаны рекомендации относительно фильтрации ICMPv6 трафика.

Следует отметить, что появление новых полей в заголовке пакета IPv6 позволяют новые правила динамического контроля потока. Например, в [9] рассматривается вариант использования поля Flow Label.

Несмотря на то, что протокол IPv6 уже существует много лет, а последние годы в его внедрении уже никто не сомневается, значительная часть современных брандмауэров еще не достигли необходимого уровня безопасности. Одной из причин является отсутствие стандартизированных механизмов защиты, которые должны присутствовать в брандмауэре по умолчанию. Именно эту проблему пытается решить группа профессионалов, которая выпустила черновой вариант стандарта с требованиями к IPv6 брандмауэрам [13]. Состоянием на июнь 2014 года выпущена вторая ревизия черновика, в котором описано, какие именно компоненты брандмауэр должен поддерживать по умолчанию и какие из них лишь следует внедрить. Авторы считают, что это поможет гарантировать, что в брандмауэре присутствует минимальный набор необходимого функционала.

IPv6 IPSec

В RFC 2460 [11] указано, что безопасность IPv6 рассматривается в стандарте, который посвящен IPsec. В текущем стандарте, посвященном требованиям, которые предъявляются к конечным узлам [14], указано, что IPsec является лишь рекомендательным средством безопасности, в отличие от старой версии этого же стандарта [15], где IPsec был обязателен для использования.

Главной трудностью реализации IPsec в IPv6 по умолчанию является создание стандартизированной глобальной инфраструктуры открытых ключей (англ. “Public Key Infrastructure, PKI”), которая смогла бы обслуживать все устройства в Интернет.

Также существуют трудности стандартизации и других областей, таких как, например, создание единой базовой конфигурации на конечных устройствах и устранение терминологической путаницы при создании подобной конфигурации (на данный момент, многие производители сетевого оборудования обладают своей специфической терминологией). Или проблема стандартизации технических моментов, например, какой механизм управления ключами использовать по умолчанию IKEv1 или IKEv2.

На данный момент уже создано множество наработок в данной сфере, к примеру, существует ряд RFC публикаций, таких как: описание IPsec PKI профилей IKEv1, IKEv2 и PKIX [16]; использование IPsec для защиты туннелей IPv6-в-IPv4 [17]; мобильные IPv6 и IKEv2 [18]. Подобных стандартов, которые относятся к IPsec-v3, можно перечислить целую массу, именно поэтому в 2011 году появилась публикация с дорожной картой, которая содержит краткие выжимки и пояснения множества различных стандартов, относящихся к IPsec и IKE [19].

Следует отметить, что в стандартном RFC, посвященном IPv6 [14], ссылка в разделе о безопасности указывает на устаревшую версию протокола IPsec (RFC 2401). В 2005 году в свет вышел RFC 4301 [19], в котором описан IPsec-v3. Ключевым изменением является то, что в новом стандарте появилось такое понятие как PAD (англ. “Peer Authentication Database”). Данная база содержит информацию, необходимую для аутентификации узла, чтобы обеспечить связь между протоколом IPsec и протоколом управления ключами (к примеру, IKE).

Также необходимо отметить, что ведутся работы и в области IKE протокола. Так, на данный момент, выпущен черновой вариант RFC публикации, в котором описан IKEv3. Данная публикация была мотивирована тем, что, несмотря на все попытки упрощения, снижения логических коллизий и несогласованностей, IKEv2 стандарт, по мнению автора, так и остался путанным и сложным.

Протокол SAVI

SAVI – технология, которая позволяет гарантировать валидность IP адресов в пределах одной локальной сети. Иными словами, SAVI создают такую инфраструктуру, которая позволяет предотвратить кражу IP адреса сторонним устройством с целью выполнения каких-либо действий от имени этого IP адреса.

SAVI работает на основе простого принципа: на устройстве, которое поддерживает SAVI (чаще всего коммутатор), создается таблица состояний, в которую записываются соответствия, какой IP адрес к какому порту привязан. Например, в FCFS (англ. “First-Come, First-Served”) SAVI такая связь создается, когда устройство впервые передает информацию в сеть через проверяющий порт (англ. “Validating Port”). Данная процедура осуществляется в несколько шагов:

1) Устройство начинает передавать данные в сеть через проверяющий SAVI порт коммутатора (на рис. 2 указан как на устройстве SAVI 1, ведущий к узлу IP 1).

2) SAVI протокол инициирует проверку: если данный IP адрес ранее не был привязан ни к одному проверяющему порту, то коммутатор (на рис. 2 коммутатор SAVI 1) сразу же создает в SAVI таблице запись соответствия “порт-IP”.

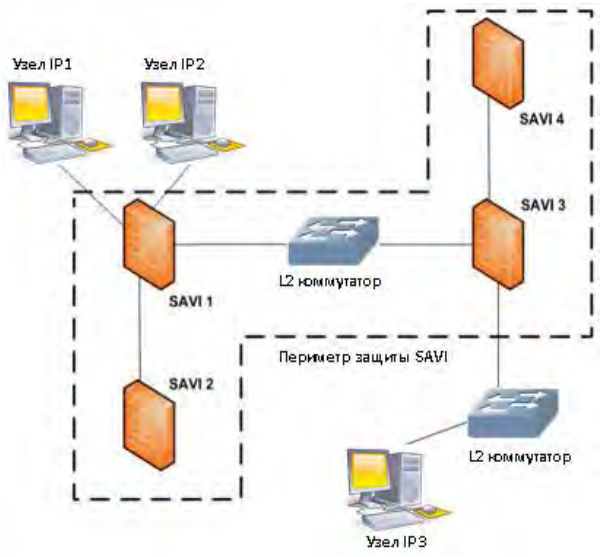

Рис. 2. Принцип работы SAVI

3) Если же этот IP адрес уже привязан к какому-либо проверяющему порту в рамках данной локальной сети (на рис. 2 границами проверки является защищенный SAVI периметр), то коммутатор инициирует проверку присутствия данного IP адреса на том проверяющем порту, на котором он был назначен ранее. Эта процедура осуществляется путем отправки в сеть запроса на проверку дублирующегося адреса (англ. “Duplicate Address Detection Network Solicitation, DAD_NS”). Если данный IP адрес более не отвечает на запросы с того проверяющего порта, к которому он был привязан ранее, то в SAVI таблице создается связка “порт-IP”.

Таким образом, то устройство, которое первым заявило, что оно обладает определенным IP адресом, и будет являться его владельцем.

Такие же механизмы были придуманы и для IP адресов, которые устройства получают динамически по DHCP протоколу [20]. Не смотря на более чем 5 летний труд и множество доработок, данный стандарт все еще находится на стадии черновика.

С SAVI связано множество уязвимостей, к примеру, если устройство покинуло сеть, то злоумышленник может забрать адрес данного устройства себе. Конечно, данный вариант не очень интересен, так как, в таком случае, злоумышленник не может организовать атаку человек-по-середине (англ. “Man-in-the-Middle, MITM”). Однако атакующий может прослушать трафик с целью дальнейшего перехвата сессии устройства, затем произвести одну из многочисленных DoS-атак на устройство жертвы и, когда жертва перестанет отвечать на запросы, перехватить сеанс связи вместе с легитимным IP-адресом. Эта и множество других атак, связанных с SAVI, описаны в [21].

Для предотвращения множества атак, связанных с перехватом и подменой IP-адресов, а также реализацией MITM, в мае 2014 выпущен в свет RFC 7219, который описывает возможность реализации SAVI с использованием SEND протокола [22].

IPv6 Routing Advertisement Guard

Еще один механизм безопасности IPv6 называется RA Guard [23]. Он призван предотвратить атаки подмены сообщений NDP протокола при обмене сообщениями между узлом и маршрутизатором сети. RA Guard является более простым и легким механизмом защиты, по сравнению с SEND протоколом, однако не является его заменителем, так как в нем отсутствуют многие механизмы безопасности присущие SEND протоколу.

RA Guard имеет два режима работы: с контролем состояния и без контроля состояния. В режиме без контроля состояния администратор должен статически указать, к какому порту L2 устройства подключен легитимный маршрутизатор (на рис. 3 порт L2 устройства, ведущий к маршрутизатору).

Рис. 3. Принцип работы RA Guard

Также администратор указывает, на основании каких критериев следует определять валидность RA сообщения от маршрутизатора:

- разрешенный/запрещенный MAC адрес;
- разрешенный/запрещенный физический порт, на который поступило RA сообщение;
- разрешенный/запрещенный IP адрес источника RA сообщения;
- разрешенный/запрещенный префикс подсети либо группа префиксов;
- приоритет маршрутизатора (англ. “Router Priority”) в соответствующем поле RA сообщения.

Порты, на которых не указано, что к ним подключен легитимный маршрутизатор, должны блокировать входящие RA сообщения (на рис. 3 порты на L2 устройстве, ведущие к хостам).

В режиме с контролем потока коммутатор анализирует сообщения, которые проходят через него в течение заданного интервала времени, либо другого тригтерного фактора (пример, количество RA сообщений). Затем коммутатор позволяет передавать RA сообщения лишь на тех портах, на которых передавались исключительно легитимные RA сообщения в течение интервала стадии анализа. Все другие порты переходят в режим блокирования входящих RA сообщений. Стоит отметить, что и при работе в данном режиме администратор должен статически указать, к какому порту L2 устройства подключен легитимный маршрутизатор, так как другого механизма определения легитимности маршрутизатора не существует.

Стандартный алгоритм RA Guard можно обойти двумя путями:

1) добавление расширенного заголовка в RA сообщение - так как многие реализации контроля RA сообщений проверяют лишь заголовок, который следует за фиксированным заголовком IPv6, вместо того, чтобы проверить всю цепочку заголовков;

2) использование фрагментации пакетов в сочетании с расширенным заголовком Destination option.

Для предотвращения данных векторов атак и улучшения RA Guard механизма предложена последовательность дополнительных проверок [24], которые коммутатор должен выполнить:

- проверить, что IP адрес сообщения не является адресом локального соединения;
- проверить, чтобы поле Hop Limit в заголовке IPv6 не было равно 255;
- проверить всю цепочку заголовков пакета и определить, не является ли пакет RA сообщением;
- проверить, не является ли полученный пакет первым фрагментом, если да, то все ли заголовки (в том числе расширенные) получены в первом фрагменте;
- при проверке заголовков проверить содержит пакет неизвестное значение в поле Next Header.

В случае, если хотя бы одна проверка не пройдена, коммутатор должен отбросить пакет, записать событие в лог-файл и классифицировать его как ошибка безопасности (англ. “security fault”).

IPv6 Snooping

Данный механизм безопасности [25] строит таблицу доверительных связей на основании ответов DHCPv6 сервера и ND сообщений. В зависимости от программного и аппаратного обеспечения каждая доверительная запись в таблице может сохранять: IPv6 адрес, MAC адрес, VLAN и порт коммутатора, на который поступило сообщение.

Основываясь на таблице доверительных записей (англ. “trusted binding table”), устройство может фильтровать сообщения, отбрасывая те, которые не имеют корректного сопоставления IPv6-MAC. Таким образом, достигается ограничение на количество IPv6 адресов, которые может объявить каждый узел и помогает предотвращать множество атак на NDP протокол.

Для повышения уровня безопасности можно заставить алгоритм проверять лишь IPv6 адреса, которые сформированы CGA протоколом, ограничить максимальное количество записей в таблице и др. Также, в случае сбоя или перезагрузки устройства алгоритм обладает функцией самовосстановления таблицы доверительных записей. При восстановлении устройство обращается к соседним узлам с просьбой предоставить свои IPv6 адреса, а также к DHCPv6 серверу с просьбой предоставления всех выданных IPv6 адресов с соответствующими им MAC адресами.

Следующий механизм безопасности **IPv6 Source/Prefix Guard** [25] неразрывно связан с IPv6 Snooping, так как использует таблицу доверительных связей для фильтрации трафика. В случае, если IPv6 адрес не находится в таблице – устройство заблокирует передачу данных от этого хоста. Также имеет возможность фильтрации трафика на основе разрешенных префиксов сетей, которые могут быть получены в RA сообщении от маршрутизатора, в сообщении делегации префикса (англ. “prefix delegation”) от DHCP, либо заданы статически. Следует отметить, что данный алгоритм не заполняет таблицу доверительных связей и может работать лишь как дополнительная функция в связке с IPv6 Snooping.

IPv6 Destination Guard

Данный алгоритм [25] работает в связке с NDP протоколом, чтобы убедиться, что процесс распознания адресов (англ. “address resolution”) производится только для работающих хостов.

Прежде, чем фильтровать трафик, алгоритм записывает в таблицу адреса активных хостов в локальной сети, прослушивая NDP и DHCP сообщения. Когда пакет достигает фильтрующего устройства, алгоритм проверяет, есть ли в таблице IPv6 адрес получателя, MAC адрес которого хочет узнать отправитель. Если в таблице соответствующая запись есть – пакет пропускается и NDP протокол продолжает процесс распознания MAC адреса удаленного хоста, в противном случае – пакет отбрасывается.

Приведенный механизм защиты может помочь в выявлении узлов, которые активно сканируют сеть на наличие работающих устройств. Каждый раз, когда хост будет обращаться к несуществующему в локальной сети IPv6 адресу, счетчик срабатываний будет возрастать, что позволит выявить нарушителя.

DHCPv6 Guard

Данный механизм безопасности [25] позволяет блокировать reply и advertisement сообщения от поддельных DHCPv6 серверов и DHCP агентов-ретрансляторов. Фильтрация происходит в зависимости от того, какая роль назначена на соответствующем порту коммутатора, транка (англ. “Trunk”) или VLAN. Также можно настроить алгоритм на более детальную сортировку, к примеру: на основе отдельных адресов и диапазона IPv6 адресов разрешенных DHCPv6 серверов и DHCP ретрансляторов, а также на основе префиксов сети.

Данный подход позволяет предотвратить атаки направленные на перенаправление трафика, и атаки типа отказ в обслуживании.

Выводы

Протокол IPv6 открыл путь для новых векторов атак в силу своей архитектуры. Это стало причиной появления новых атак, методы борьбы с которыми не существовали в прошлой версии протокола IP. Для борьбы с явными угрозами разработаны механизмы безопасности, значительная часть из которых основано на применении криптографии. Введение некоторых из предложенных механизмов обеспечения безопасности

позволит достичь лишь частичной защиты. Только использование комплексных методов защиты позволят создать периметр надлежащего уровня защищенности. Однако не стоит забывать, что сами механизмы безопасности также открыли новые потенциальные векторы атак, в основном по средствам реализации DoS атак на сложные для вычисления операции.

Следует отметить, что, несмотря на то, что IPsec еще не может быть внедрен в IPv6 протокол как стандарт обеспечения безопасности "из коробки", все же его можно задействовать вручную. Данный протокол позволяет обеспечить две важные составляющие безопасности данных: конфиденциальность и целостность. Однако IPsec не гарантирует доступность информации.

Учитывая то, что большинство защитных механизмов подвержены DoS атакам разного рода, необходима разработка новых эффективных методов защиты от данных угроз.

Список литературы

1. Aura T. Cryptographically generated addresses (CGA) / T. Aura. – 2005.

2. Bagnulo M. Efficient security for IPv6 multihoming / M. Bagnulo, A. García-Martínez, A. Azcorra //ACM SIGCOMM Computer Communication Review. – 2005. – T. 35, №. 2. – P. 61-68.

3. Bos J.W. Analysis and optimization of cryptographically generated addresses / J.W. Bos, O. Özen, J.P. Hubaux // Information Security. – Springer Berlin Heidelberg, 2009. – P. 17-32.

4. Bagnulo M. Cryptographically Generated Addresses (CGA) Extension Field Format / M. Bagnulo, J. Arkko // RFC 4581, October, 2006.

5. Arkko J. Secure neighbor discovery (SEND) / J. Arkko, Ed. Ericsson, J. Kempf // RFC 3971, March, 2005.

6. CGA as alternative security credentials with IKEv2: implementation and analysis / J.M. Combes et al. // SAR-SSI'12: 7th Conference on Network Architectures and Information Systems Security. – 2012. – P. 53-59.

7. Bagnulo M. Hash-based addresses (HBA) / M. Bagnulo. – 2009.

8. Davies E. IPv6 transition/co-existence security considerations / E. Davies, S. Krishnan, P. Savola. – 2007.

9. McGann O. IPv6 packet filtering: дис. / O. McGann. – National University of Ireland Maynooth, 2005.

10. Krishnan S. Handling of Overlapping IPv6 Fragments / S. Krishnan. – 2009.

11. Deering S.E. Internet protocol, version 6 (IPv6) specification / S.E. Deering. – 1998.

12. Davies E. Recommendations for filtering icmpv6 messages in firewalls / E. Davies, J. Mohacsi // RFC 4890, May, 2007.

13. Gont F. Requirements for IPv6 Enterprise Firewalls / F. Gont, M. Ermini, W. Liu. – April, 2014.

14. Jankiewicz E. IPv6 Node Requirements / E. Jankiewicz, J. Loughney, T. Narten // RFC 6434, December 2011.

15. Loughney J. IPv6 node requirements / J. Loughney. – 2006.

16. Korver B. The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX / B. Korver. – 2007.

17. Using IPsec to Secure IPv6-in-IPv4 Tunnels / R. Graveman . et al. // RFC4891, May. – 2007.

18. Devarapalli V. Mobile IPv6 operation with IKEv2 and the revised IPsec architecture / V. Devarapalli, F. Dupont. – 2007.

19. Frankel S. IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap / S. Frankel, S. Krishnan // RFC 6071, February, 2011.

20. SAVI Solution for DHCP / J. Bi et al. – May, 2014.

21. McPherson D. Source Address Validation Improvement (SAVI) Threat Scope / D. McPherson, J. Halpern, F. Baker. – 2013.

22. Bagnulo M. SEcure Neighbor Discovery (SEND) Source Address Validation Improvement (SAVI) / M. Bagnulo, A. Garcia-Martinez. – 2014.

23. IPv6 Router Advertisement Guard / J. Mohacsi et al. – 2011.

24. Gont F. Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard) / F. Gont. – 2014.

25. IPv6 First-Hop Security Configuration Guide, Cisco IOS Release 15S [Электронный ресурс]: Cisco Systems. – 26 Nov 2012. – Режим доступа : http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_fhsec/configuration/15-s/ip6f-15-s-book.pdf.

Поступила в редколлегию 16.12.2014

Рецензент: д-р техн. наук, проф. А.В. Лемешко, Харьковский национальный университет радиоэлектроники, Харьков.

МЕХАНІЗМИ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ СТЕКА ПРОТОКОЛІВ IPV6

А.В. Снігуров, В.Х. Чакрян

В статті наведено аналіз механізмів забезпечення безпеки сімейства протоколів IPv6, таких як: SEND, IPv6 Firewall, IPv6 IPSec, SAVI, IPv6 Routing Advertisement Guard, IPv6 Source/Prefix Guard, IPv6 Destination Guard, IPv6 Snooping, DHCPv6 Guard. В статті проаналізовано поточний стан та невирішені питання досліджуваної області. Зроблено акцент на те, що дані механізми безпеки, в деяких випадках, створюють нові потенційні вектори атак. Результати статті можуть бути використані для розробки нових механізмів забезпечення інформаційної безпеки телекомунікаційних мереж.

Ключові слова: *IPv6, механізми безпеки IPv6, SEND, CGA, HBA, SAVI, IPv6 IPSec, IPv6 Firewall, IPv6 First Hop Security.*

SECURITY MECHANISMS OF IPV6 PROTOCOL FAMILY

A.V. Snigurov, V.K. Chakrian

In this paper the security mechanisms of IPv6 protocol family are analyzed, among them: SEND, IPv6 Firewall, IPv6 IPSec, SAVI, IPv6 Routing Advertisement Guard, IPv6 Source/Prefix Guard, IPv6 Destination Guard, IPv6 Snooping, DHCPv6 Guard. In the paper it's analyzed the current state and unresolved questions of research field. The paper is focused on the fact, that in some circumstances, these security mechanisms can create new potential vectors of attack. The results of the paper can used to create the new information security mechanisms in telecommunication systems.

Keywords*: IPv6, security mechanisms of IPv6, SEND, CGA, HBA, SAVI, IPv6 IPSec, IPv6 Firewall, IPv6 First Hop Security.*