Лекция №19. ПРОТОКОЛЫ ЗАЩИТЫ НА КАНАЛЬНОМ И СЕАНСОВОМ УРОВНЯХ

Оглавление

[1. Модель взаимодействия систем ISO/OSI и стек протоколов TCP/IP 1](#_Toc509029641)

[1.1.Структура и функциональность стека протоколов TCP/IP 2](#_Toc509029642)

[2. Протоколы формирования защищенных каналов на канальном уровне 6](#_Toc509029643)

[2.1. Протокол РРТР 7](#_Toc509029644)

[2.2. Протокол L2TP 9](#_Toc509029645)

[3. Протоколы формирования защищенных каналов на сеансовом уровне 13](#_Toc509029646)

[3.1. Протоколы SSL и TLS 13](#_Toc509029647)

[4. Защита беспроводных сетей 16](#_Toc509029648)

Базовые технологии сетевой безопасности построены на протоколах защиты на канальном и сеансовом уровнях модели ISO/OSI. Рассмотрим модель взаимодействия систем ISO/OSI и стек протоколов TCP/IP для понимания принципов защиты информации в сетях.

# Модель взаимодействия систем ISO/OSI и стек протоколов TCP/IP

Модель ISO/OSI определяет различные уровни взаимодействия систем и указывает, какие функции должен выполнять каждый уровень.

В модели OSI средства взаимодействия делятся на семь уровней: прикладной (Application), представительный (Presentation), сеансовый (Session), транспортный (Transport), сетевой (Network), канальный (Data Link) и физический (Physical). Самый верхний уровень – прикладной. На этом уровне пользователь взаимодействует с приложениями. Самый нижний уровень – физический. Этот уровень обеспечивает обмен сигналами между устройствами.

Обмен данными через каналы связи происходит путем перемещения данных с верхнего уровня на нижний, затем транспортировки по линиям связи и, наконец, обратным воспроизведением данных в компьютере клиента в результате их перемещения с нижнего уровня на верхний.

Для обеспечения необходимой совместимости на каждом из уровней архитектуры компьютерной сети действуют *специальные стандартные протоколы*. Они представляют собой формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты, лежащие на одном уровне, но в разных узлах сети.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется *стеком коммуникационных протоколов*. Следует четко различать модель ISO/OSI и стек протоколов ISO/OSI. *Модель* ISO/OSI является концептуальной схемой взаимодействия открытых систем, а *стек протоколов* ISO/OSI представляет собой набор вполне конкретных спецификаций протоколов для семи уровней взаимодействия, которые определены в модели ISO/OSI.

Коммуникационные протоколы могут быть реализованы как программно, так и аппаратно. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней – как правило, чисто программными средствами.

Модули, реализующие протоколы соседних уровней и находящиеся в одном узле сети, должны взаимодействовать друг с другом также в соответствии с четко определенными правилами и с помощью стандартизованных форматов сообщений. Эти правила принято называть *межуровневым интерфейсом.* Межуровневый интерфейс определяет набор сервисов, предоставляемых данным уровнем соседнему уровню. В сущности, протокол и интерфейс являются близкими понятиями, но традиционно в сетях за ними закреплены разные области действия: *протоколы* определяют правила взаимодействия модулей одного уровня в разных узлах сети, а *интерфейсы* определяют правила взаимодействия модулей соседних уровней в одном узле.

*Стек протоколов TCP/IP* (*Transmission Control Protocol/Internet Protocol*) является промышленным стандартом стека коммуникационных протоколов, разработанным для глобальных сетей. Стандарты TCP/IP опубликованы в серии документов, названных Request for Comment (RFC).

Стек ТСР/IР объединяет в себе целый набор взаимодействующих между собой протоколов. Самыми важными из них являются протокол ІР, отвечающий за поиск маршрута (или маршрутов) в Интернете от одного компьютера к другому через множество промежуточных сетей, шлюзов и маршрутизаторов и передачу блоков данных по этим маршрутам, и протокол ТСР, обеспечивающий надежную доставку, безошибочность и правильный порядок приема передаваемых данных. Большой вклад в развитие стека TCP/IP внес Калифорнийский университет в Беркли (США), который реализовал протоколы стека в своей версии OC UNIX, сделав как сами программы, так и их исходные тексты общедоступными и бесплатными. Популярность этой операционной системы привела к широкому распространению IP, TCP и других протоколов стека. Сегодня этот стек используется для связи компьютеров Всемирной информационной сети Интернет, а также в огромном числе корпоративных сетей. Стек TCP/IP является самым распространенным средством организации составных компьютерных сетей.

Широкое распространение стека TCP/IP объясняется следующими его свойствами:

* + - наиболее завершенный стандартный и в то же время популярный стек сетевых протоколов, имеющий многолетнюю историю;
		- почти все большие сети передают основную часть своего трафика с помощью протокола TCP/IP;
		- все современные операционные системы поддерживают стек TCP/IP;
		- метод получения доступа к сети Интернет;
		- гибкая технология для соединения разнородных систем на уровне как транспортных подсистем, так и прикладных сервисов;
		- основа для создания корпоративной интранет-сети, использующей транспортные услуги Интернета и гипертекстовую технологию WWW, разработанную в Интернете;
		- устойчивая масштабируемая межплатформенная среда для приложений клиент/сервер.

## 1.1.Структура и функциональность стека протоколов TCP/IP

****Стек TCP/IP был разработан до появления модели взаимодействия открытых систем OSI и также имеет многоуровневую структуру. Структура протоколов TCP/IP приведена на рис. 1.

***Рис. 1. Уровни стека протоколов TCP/IP***

Стек протоколов TCP/IP имеет четыре уровня: прикладной (Application), транспортный (Transport), уровень межсетевого взаимодействия (Internet) и уровень сетевых интерфейсов (Network). Для сравнения на рис. 1 показаны также семь уровней модели OSI. Следует отметить, что соответствие уровней стека TCP/IP уровням модели OSI достаточно условно.

***Прикладной уровень* (*Application*)** включает большое число прикладных протоколов и сервисов. К ним относятся такие популярные протоколы, как протокол копирования файлов FTP, протокол эмуляции терминала Telnet, почтовый протокол SMTP, используемый в электронной почте сети Интернет, гипертекстовые сервисы доступа к удаленной информации, например, WWW, и многие другие. Рассмотрим несколько подробнее некоторые из этих протоколов. *Протокол пересылки файлов FTP* (*File Transfer Protocol*) реализует удаленный доступ к файлу. Для того чтобы обеспечить надежную передачу, FTP использует в качестве транспорта протокол с установлением соединений – TCP. Кроме пересылки файлов, протокол FTP предлагает и другие услуги. Например, пользователю предоставляется возможность интерактивной работы с удаленной машиной, в частности он может распечатать содержимое ее каталогов. Наконец, FTP выполняет аутентификацию пользователей. Прежде чем получить доступ к файлу, в соответствии с протоколом пользователи должны сообщить свое имя и пароль. Для доступа к публичным каталогам FTP-архивов Интернета не требуется парольная аутентификация, и ее можно обойти путем использования для такого доступа предопределенного имени пользователя Anonymous.

В стеке TCP/IP протокол FTP предлагает наиболее широкий набор услуг для работы с файлами, однако он является и самым сложным для программирования. Если приложению не требуются все возможности протокола FTP, тогда можно использовать простой протокол пересылки файлов TFTP (Trivial File Transfer Protocol). Этот протокол реализует только передачу файлов, причем в качестве транспорта используется протокол без установления соединения – UDP.

*Протокол Telnet* обеспечивает передачу потока байтов между процессами, а также между процессом и терминалом. Наиболее часто этот протокол используется для эмуляции терминала удаленного компьютера. При использовании сервиса Telnet пользователь фактически управляет удаленным компьютером, так же как и локальный пользователь, поэтому подобный вид доступа требует хорошей защиты. Серверы Telnet всегда используют как минимум аутентификацию по паролю, а иногда и более мощные средства защиты, например, систему Kerberos.

*Протокол SNMP* (*Simple Network Management Protocol*) используется для организации сетевого управления. Сначала протокол SNMP был разработан для удаленного контроля и управления маршрутизаторами Интернета. С ростом популярности протокол SNMP стали применять для управления разным коммуникационным оборудованием – концентраторами, мостами, сетевыми адаптерами и др. В стандарте SNMP определена спецификация информационной базы

данных управления сетью. Эта спецификация, известная как база данных MIB (Management Information Base), определяет те элементы данных, которые управляемое устройство должно сохранять, и допустимые операции над ними.

На ***транспортном уровне* (*Transport*)** стека TCP/IP, называемом также основным уровнем, функционируют протоколы TCP и UDP.

*Протокол управления передачей ТС* (*Transmission Control Protocol*) решает задачу обеспечения надежной информационной связи между двумя конечными узлами. Этот протокол называют протоколом «с установлением соединения». Это означает, что два узла, связывающиеся при помощи этого протокола, «договариваются» о том, что они будут обмениваться потоком данных, и принимают некоторые соглашения об управлении этим потоком. Согласно протоколу TСР, отправляемые данные «нарезаются» на небольшие стандартные пакеты, после чего каждый пакет маркируется таким образом, чтобы в нем были данные для правильной сборки документа на компьютере получателя.

*Протокол дейтаграмм пользователя UDP* (*User Datagram Protocol*) обеспечивает передачу прикладных пакетов дейтаграммным способом, то есть каждый блок передаваемой информации (пакет) обрабатывается и распространяется от узла к узлу как независимая единица информации – дейтаграмма. При этом протокол UDР выполняет только функции связующего звена между сетевым протоколом и многочисленными прикладными процессами. Необходимость в протоколе UDР обусловлена тем, что он «умеет» различать приложения и доставляет информацию от приложения к приложению.

***Уровень межсетевого взаимодействия* (*Internet*)** реализует концепцию коммутации пакетов без установления соединений. Основным протоколом этого уровня является *адресный протокол IP*. Этот протокол изначально проектировался как протокол передачи пакетов в составных сетях, которые состоят из большого количества локальных сетей, объединенных как локальными, так и глобальными связями.

Суть протокола IP заключается в том, что у каждого пользователя Всемирной сети Интернет должен быть свой уникальный адрес (IP-адрес). Без этого нельзя говорить о точной доставке ТСР-пакетов в нужное место. Этот адрес выражается очень просто – четырьмя байтами, например, 185.47.39.14. Структура IP-адреса организована таким образом, что каждый компьютер, через который проходит какой-либо ТСР-пакет, может по этим четырем числам определить, кому из ближайших «соседей» надо переслать пакет, чтобы он оказался «ближе» к получателю. В результате конечного числа перебросок ТСР-пакет достигает адресата. В данном случае оценивается не географическая близость. В расчет принимаются условия связи и пропускная способность линии. Два компьютера, находящиеся на разных континентах, но связанные высокопроизводительной линией космической связи, считаются более близкими друг другу, чем два компьютера из соседних городов, связанных обычной телефонной связью. Решением вопросов, что считать «ближе», а что «дальше», занимаются специальные средства – маршрутизаторы. Роль маршрутизатора в сети может выполнять как специализированный компьютер, так и специализированная программа, работающая на узловом сервере сети.

К уровню межсетевого взаимодействия относятся и протоколы, связанные с составлением и модификацией таблиц маршрутизации, такие как *протоколы сбора маршрутной информации RIP* (*Routing Internet Protocol*) и *OSPF* (*Open Shortest Path First*), а также *протокол межсетевых управляющих сообщений ICMP* (*Internet Control Message Protocol*). Последний протокол предназначен для обмена информацией об ошибках между маршрутизаторами сети и узлом-источником пакета.

***Уровень сетевого интерфейса* (*Network*)** соответствует физическому и канальному уровням модели OSI. Этот уровень в протоколах TCP/IP не регламентируется, но поддерживает все популярные стандарты физического и канального уровня: для локальных сетей это Ethernet, Token Ring, FDDI, Fast Ethernet, для глобальных сетей – протоколы соединений точка–точка SLIP и PPP, протоколы территориальных сетей с коммутацией пакетов X.25, Frame Relay. Разработана спецификация, определяющая использование технологии ATM в качестве транспорта канального уровня.

Разделенные на уровни протоколы стека TCP/IP спроектированы таким образом, что конкретный уровень хоста назначения получает именно тот объект, который был отправлен эквивалентным уровнем хоста источника. Каждый уровень стека одного хоста образует логическое соединение с одноименным уровнем стека другого хоста. При реализации физического соединения уровень передает свои данные интерфейсу уровня, расположенного выше или ниже в том же хосте. На рис. 2 показано, как осуществляются физическое и логическое соединения уровней. Вертикальные стрелки показывают физическое соединение в рамках одного хоста, а горизонтальные – логическое соединение между одноименными уровнями в различных хостах.



***Рис. 2. Логические и физические соединения между уровнями стека TCP/IP***

Следует обратить внимание на терминологию, традиционно используемую для обозначения информационных объектов, которые распространяются на интерфейсах между различными уровнями управления стека протоколов TCP/IP.

Приложение передает транспортному уровню сообщение (message), которое имеет соответствующее данному приложению размер и семантику. Транспортный уровень «разрезает» это сообщение (если оно достаточно велико) на пакеты (packets), которые передаются уровню межсетевого взаимодействия (то есть протоколу IP). Протокол IP формирует свои IP-пакеты (еще говорят «IP-дейтаграммы») и затем упаковывает их в формат, приемлемый для данной физической среды передачи информации. Эти, уже аппаратно-зависимые, пакеты обычно называют кадрами (Frame).

Когда данные передаются от прикладного уровня к транспортному, затем к уровню межсетевого взаимодействия и далее через уровень сетевого интерфейса в сеть, каждый протокол выполняет соответствующую обработку и инкапсулирует результат этой обработки, присоединяя спереди свой заголовок. На рис. 3 показана схема процесса инкапсуляции передаваемых данных и формирования заголовков пакетов в стеке TCP/IP.

***Рис. 3. Схема инкапсуляции данных в стеке протоколов TCP/IP***

В системе, принимающей данный поток информации, эти заголовки последовательно удаляются по мере обработки данных и передачи их вверх по стеку. Такой подход обеспечивает необходимую гибкость в обработке передаваемых данных, поскольку верхним уровням вовсе не нужно касаться технологии, используемой на нижних уровнях. Например, если шифруются данные на уровне IP, уровень TCP и прикладной остаются неизменными.

Что касается безопасности протоколов ТСР/ІР, то есть безопасности передачи данных в Интернете в целом, пользователям необходимо иметь в виду, что, если не принято специальных мер, все данные передаются протоколами ТСР/ІР в открытом виде. Это значит, что любой узел (и соответственно, его оператор), находящийся на пути следования данных от отправителя к получателю, может скопировать себе все передаваемые данные и использовать их в дальнейшем в своих целях. В равной мере данные могут быть искажены или уничтожены. Виртуальный защищенный канал можно построить с помощью системных средств, реализованных на разных уровнях модели взаимодействия открытых систем OSI. От выбранного рабочего уровня OSI зависят функциональность реализуемого виртуального защищенного канала и его совместимость с приложениями корпоративной информационной системы, а также с другими средствами защиты.

Для независимости от прикладных протоколов и приложений виртуальные защищенные каналы формируются на одном из более низких уровней модели OSI – канальном, сетевом или сеансовом.

Средства, применяемые на *канальном уровне* модели OSI, позволяют обеспечить инкапсуляцию различных видов трафика третьего уровня (и более высоких) и построение виртуальных туннелей типа точка–точка (от маршрутизатора к маршрутизатору или от персонального компьютера к шлюзу ЛВС).

При построении защищенных виртуальных каналов на *сеансовом уровне* появляется возможность криптографической защиты информационного обмена, включая аутентификацию, а также реализации ряда функций посредничества между взаимодействующими сторонами.

# Протоколы формирования защищенных каналов на канальном уровне

Протоколы PPTP (Point-to-Point Tunneling Protocol) и L2TP (Layer-2 Tunneling Protocol) являются протоколами туннелирования канального уровня модели OSI. Общим свойством этих протоколов является то, что они используются для организации защищенного многопротокольного удаленного доступа к ресурсам корпоративной сети через открытую сеть, например, через Интернет.

Оба протокола – PPTP и L2TP – обычно относят к протоколам формирования защищенного канала, однако этому определению точно соответствует только протокол РРТР, который обеспечивает туннелирование и шифрование передаваемых данных. Протокол L2TP является протоколом туннелирования, поскольку поддерживает только функции туннелирования. Функции защиты данных (шифрование, целостность, аутентификация) в этом протоколе не поддерживаются. Для защиты туннелируемых данных в протоколе L2TP необходимо использовать дополнительный протокол, в частности IPSec.

Клиентское программное обеспечение обычно использует для удаленного доступа стандартный протокол канального уровня РРР (Point-to-Point Protocol). Протоколы PPTP и L2TP основываются на протоколе PPP и являются его расширениями. Первоначально протокол PPP, расположенный на канальном уровне, был разработан для инкапсуляции данных и их доставки по соединениям типа точка– точка. Этот протокол служит также для организации асинхронных (например, коммутируемых) соединений.

В набор PPP входят протокол управления соединением LCP (Link Control Protocol), ответственный за конфигурацию, установку, работу и завершение соединения точка–точка, и протокол управления сетью NCP (Network Control Protocol), способный инкапсулировать в PPP протоколы сетевого уровня для транспортировки через соединение точка–точка. Это позволяет одновременно передавать пакеты Novell IPX и Microsoft IP по одному соединению PPP.

Для доставки конфиденциальных данных из одной точки в другую через сети общего пользования сначала производится инкапсуляция данных с помощью протокола PPP, затем протоколы PPTP и L2TP выполняют шифрование данных и собственную инкапсуляцию. После того как туннельный протокол доставляет пакеты из начальной точки туннеля в конечную, выполняется деинкапсуляция.

На физическом и канальном уровнях протоколы PPTP и L2TP идентичны, но на этом их сходство заканчивается и начинаются различия.

## 2.1. Протокол РРТР

Протокол РРТР (Point-to-Point Tunneling Protocol), разработанный компанией Майкрософт при поддержке ряда других компаний, предназначен для создания защищенных виртуальных каналов при доступе удаленных пользователей к локальным сетям через Интернет. Протокол РРТР предполагает создание криптозащищенного туннеля на канальном уровне модели OSI для случаев как прямого соединения удаленного компьютера с открытой сетью, так и подсоединения его к открытой сети по телефонной линии через провайдера.

Протокол РРТР получил практическое распространение благодаря компании Майкрософт, реализовавшей его в своих операционных системах Windows NT/2000. Некоторые производители межсетевых экранов и шлюзов VPN также поддерживают протокол РРТР. Протокол РРТР позволяет создавать защищенные каналы для обмена данными по протоколам IP, IPX или NetBEUI. Данные этих протоколов упаковываются в кадры РРР и затем инкапсулируются посредством протокола РРТР в пакеты протокола IP, с помощью которого переносятся в зашифрованном виде через любую сеть ТСР/IP.

Пакеты, передаваемые в рамках сессии РРТР, имеют следующую структуру (рис. 4):

***Рис. 4. Структура пакета для пересылки по туннелю PPTP***

* + - заголовок канального уровня, используемый внутри Интернета, например, заголовок кадра Ethernet;
		- заголовок IP, содержащий адреса отправителя и получателя пакета;
		- заголовок общего метода инкапсуляции для маршрутизации GRE (Generic Routing Encapsulation);
		- исходный пакет РРР, включающий пакет IP, IPX или NetBEUI.

Принимающий узел сети извлекает из пакетов IP кадры РРР, а затем извлекает из кадра РРР исходный пакет IP, IPX или NetBEUI и отправляет его по локальной сети конкретному адресату. Многопротокольность инкапсулирующих протоколов канального уровня, к которым относится протокол РРТР, является их важным преимуществом перед протоколами защищенного канала более высоких уровней. Например, если в корпоративной сети используются IPX или NetBEUI, применение протоколов IPSec или SSL просто невозможно, поскольку они ориентированы только на один протокол сетевого уровня IP.

Данный способ инкапсуляции обеспечивает независимость от протоколов сетевого уровня модели OSI и позволяет осуществлять защищенный удаленный доступ через открытые IP-сети к любым локальным сетям (IP, IPX или NetBEUI). Согласно протоколу РРТР, при создании защищенного виртуального канала производятся аутентификация удаленного пользователя и шифрование передаваемых данных (рис. 5).

***Рис. 5. Архитектура протокола РРТР***

Для аутентификации удаленного пользователя могут применяться различные протоколы, используемые для PPP. В реализации РРТР, включенной компанией Майкрософт в Windows NT/2000, поддерживаются следующие протоколы аутентификации: протокол аутентификации по паролю РАР (Password Authentication Protocol), протокол аутентификации при рукопожатии MSCHAP (Microsoft ChallengeHandshaking Authentication Protocol) и протокол аутентификации EAP-TLS (Extensible Authentication Protocol-Transport Layer Security). При использовании протокола РАР идентификаторы и пароли передаются по линии связи в незашифрованном виде, при этом только сервер проводит аутентификацию клиента. При использовании протоколов MSСНАР и EAP-TLS обеспечиваются защита от повторного использования злоумышленником перехваченных пакетов с зашифрованным паролем и взаимная аутентификация клиента и VPN-сервера.

Шифрование с помощью PPTP гарантирует, что никто не сможет получить доступ к данным при пересылке через Интернет. Шифрование MPPE (Microsoft Point-to-Point Encryption) совместимо только с MSCHAP (версии 1 и 2) и EAP-TLS и умеет автоматически выбирать длину ключа шифрования при согласовании параметров между клиентом и сервером. Шифрование MPPE поддерживает работу с ключами длиной 40, 56 или 128 бит. Протокол PPTP изменяет значение ключа шифрования после каждого принятого пакета.

Протокол РРТР применяется в схеме туннелирования при прямом подсоединении компьютера удаленного пользователя к Интернету. Рассмотрим реализацию этой схемы туннелирования (рис. 6).

***Рис. 6. Схема туннелирования при прямом подсоединении компьютера удаленного пользователя к Интернету***

Удаленный пользователь устанавливает удаленное соединение с локальной сетью с помощью клиентской части сервиса удаленного доступа RAS (Remote Access Service), входящего в состав Windows. Затем пользователь обращается к серверу удаленного доступа локальной сети, указывая его IP-адрес, и устанавливает с ним связь по протоколу РРТР. Функции сервера удаленного доступа может выполнять пограничный маршрутизатор локальной сети.

На компьютере удаленного пользователя должны быть установлены клиентская часть сервиса RAS и драйвер РРТР, которые входят в состав Windows 98/NT, а на сервере удаленного доступа локальной сети – сервер RAS и драйвер РРТР, входящие в состав Windows NT Server. Протокол РРТР определяет несколько служебных сообщений, которыми обмениваются взаимодействующие стороны. Служебные сообщения передаются по протоколу ТСР.

После успешной аутентификации начинается процесс защищенного информационного обмена. Внутренние серверы локальной сети могут не поддерживать протокола РРТР, поскольку пограничный маршрутизатор извлекает кадры РРР из пакетов IP и посылает их по локальной сети в необходимом формате – IP, IPX или NetBIOS.

## 2.2. Протокол L2TP

Протокол L2TP (Layer-2 Tunneling Protocol) разработан в организации IETF (Internet Engineering Task Force) при поддержке компаний Майкрософт и Cisco Systems. Протокол L2TP разрабатывался как протокол защищенного туннелирования РРР-трафика через сети общего назначения с произвольной средой.

В отличие от РРТР, протокол L2TP не привязан к протоколу IP, поэтому он может быть использован в сетях с коммутацией пакетов, например в сетях АТМ (Asynchronous Transfer Mode) или в сетях с ретрансляцией кадров (Frame Relay). Кроме того, в протокол L2TP добавлена важная функция управления потоками данных.

В протокол L2TP также добавлены ряд отсутствующих в спецификации протокола РРТР функций защиты, в частности включена возможность работы с протоколами АН и ESP стека протоколов IPSec. Архитектура протокола L2TP представлена на рис. 7.

***Рис. 7. Архитектура протокола L2TP***

Протоколы АН и ESP являются основными компонентами стека протоколов IPSeс. Эти протоколы допускают использование пользователями по их согласованному выбору различных криптографических алгоритмов шифрования и аутентификации. На домен интерпретации DOI (Domain of Interpretation) возложены функции обеспечения совместной работы используемых протоколов и алгоритмов.

В сущности, гибридный протокол L2TP представляет собой расширение протокола РРР функциями аутентификации удаленных пользователей, создания защищенного виртуального соединения и управления потоками данных.

Протокол L2TP применяет в качестве транспорта протокол UDP и использует одинаковый формат сообщений как для управления туннелем, так и для пересылки данных. В реализации Майкрософт протокол L2TP использует в качестве контрольных сообщений пакеты UDP, содержащие шифрованные пакеты PPP. Надежность доставки гарантирует контроль последовательности пакетов.

Как и в случае с PPTP, протокол L2TP начинает сборку пакета для передачи в туннель с того, что к полю информационных данных PPP добавляется сначала заголовок PPP, а затем заголовок L2TP. Полученный таким образом пакет инкапсулируется протоколом UDP. В качестве порта отправителя и получателя протокол L2TP использует UDP-порт 1701. На рис. 8.8 показана структура пакета для пересылки по туннелю L2TP.

***Рис. 8.8. Структура пакета для пересылки по туннелю L2TP***

В зависимости от выбранного типа политики безопасности стека протоколов IPSec протокол L2TP может шифровать UDP-сообщения и добавлять к ним заголовок и окончание ESP (Encapsulating Security Payload), а также окончание IPSec ESP Authentication. Затем производится инкапсуляция в IP. Добавляется IP-заголовок, содержащий адреса отправителя и получателя. В завершение L2TP выполняет вторую PPP-инкапсуляцию для подготовки данных к передаче.

Компьютер-получатель принимает данные, обрабатывает заголовок и окончание PPP, убирает заголовок IP. При помощи IPSec ESP Authentication проводится аутентификация информационного поля IP, а протокол ESP IPSec помогает расшифровать пакет. Далее компьютер обрабатывает заголовок UDP и использует заголовок L2TP для идентификации туннеля. Теперь пакет PPP содержит только полезные данные, которые обрабатываются или пересылаются указанному получателю.

Хотя протокол PPTP обеспечивает достаточную степень безопасности, но все же протокол L2TP (поверх IPSec) надежнее. Протокол L2TP поверх IPSec обеспечивает аутентификацию на уровнях «пользователь» и «компьютер», а также выполняет аутентификацию и шифрование данных. На первом этапе аутентификации клиентов и серверов VPN протокол L2TP использует локальные сертификаты, полученные от службы сертификации. Клиент и сервер обмениваются сертификатами и создают защищенное соединение ESP SA (Security Association).

После того как L2TP (поверх IPSec) завершает процесс аутентификации компьютера, выполняется аутентификация на уровне пользователя. Для этой аутентификации можно задействовать любой протокол, даже PAP, передающий имя пользователя и пароль в открытом виде. Это вполне безопасно, так как L2TP (поверх IPSec) шифрует всю сессию. Однако проведение аутентификации пользователя при помощи MSCHAP, применяющего различные ключи шифрования для аутентификации компьютера и пользователя, может повысить безопасность.

Протокол L2TP предполагает использование схемы, в которой туннель образуется между сервером удаленного доступа провайдера и маршрутизатором корпоративной сети. В отличие от своих предшественников – протоколов РРТР и L2F, – протокол L2TP предоставляет возможность открывать между конечными абонентами сразу несколько туннелей, каждый из которых может быть выделен для отдельного приложения. Эти особенности обеспечивают гибкость и безопасность туннелирования.

Согласно спецификации протокола L2TP, роль сервера удаленного доступа провайдера должен выполнять концентратор доступа LAC (L2TP Access Concentrator), который реализует клиентскую часть протокола L2TP и обеспечивает удаленному пользователю сетевой доступ к его локальной сети через Интернет. В качестве сервера удаленного доступа локальной сети должен выступать сетевой сервер LNS (L2TP Network Server), функционирующий на совместимых с протоколом РРР платформах (рис. 9).

***Рис. 9. Схемы туннелирования по протоколу L2TP***

Аналогично протоколам РРТР и L2F, формирование защищенного виртуального канала в протоколе L2TP осуществляется в три этапа:

* + - установление соединения с сервером удаленного доступа локальной сети;
		- аутентификация пользователя;
		- конфигурирование защищенного туннеля .

На первом этапе для установления соединения с сервером удаленного доступа локальной сети удаленный пользователь инициирует РРР-соединение с провайдером ISP. Концентратор доступа LAC, функционирующий на сервере провайдера ISP, принимает это соединение и устанавливает канал РРР. Затем концентратор доступа LAC выполняет частичную аутентификацию конечного узла и его пользователя. Используя только имя пользователя, провайдер ISP решает, нужен ли пользователю сервис туннелирования L2TP. Если такой сервис нужен, то следующим шагом для концентратора доступа LAC будет выяснение адреса сетевого сервера LNS, с которым нужно установить туннельное соединение. Для удобства определения соответствия между пользователем и сервером LNS, обслуживающим сеть

пользователя, может использоваться база данных, поддерживаемая провайдером ISP для своих клиентов.

После выяснения IP-адреса сервера LNS производится проверка, не существует ли уже туннель L2TP с этим сервером. Если такого туннеля нет, то он устанавливается. Между концентратором доступа провайдера LAC и сетевым сервером LNS локальной сети устанавливается сессия по протоколу L2TP.

При создании туннеля между LAC и LNS новому соединению в рамках этого туннеля присваивается идентификатор, называемый идентификатором вызова Сall ID. Концентратор LAC отправляет сетевому серверу LNS пакет с уведомлением о вызове с данным Сall ID. Сервер LNS может принять этот вызов или отклонить его.

На втором этапе после установления сессии L2TP сетевой сервер LNS локальной сети выполняет процесс аутентификации пользователя. Для этого может быть использован один из стандартных алгоритмов аутентификации, в частности СНАР. В случае применения протокола аутентификации СНАР пакет уведомления включает слово-вызов, имя пользователя и его ответ. Для протокола РАР эта информация состоит из имени пользователя и незашифрованного пароля. Сетевой сервер LNS может сразу использовать эту информацию для выполнения аутентификации, чтобы не заставлять удаленного пользователя повторно вводить свои данные и не осуществлять дополнительного цикла аутентификации.

При отправке результата аутентификации сетевой сервер LNS может также передать концентратору доступа LAC сведения об IPадресе узла пользователя. По существу, коцентратор доступа LAC работает как посредник между узлом удаленного пользователя и сетевым сервером LNS локальной сети.

На третьем этапе в случае успешной аутентификации пользователя создается защищенный туннель между концентратором доступа LAC провайдера и сервером LNS локальной сети. В результате инкапсулированные кадры РРР могут передаваться по туннелю между концентратором LAC и сетевым сервером LNS в обоих направлениях. При поступлении кадра РРР от удаленного пользователя концентратор LAC удаляет из него байты обрамления кадра, байты контрольной суммы, затем инкапсулирует его с помощью протокола L2TP в сетевой протокол и отправляет по туннелю сетевому серверу LNS. Сервер LNS, используя протокол L2TP, извлекает из прибывшего пакета кадр РРР и обрабатывает его стандартным образом.

Настройка необходимых значений параметров туннеля производится с помощью управляющих сообщений. Протокол L2TP может работать поверх любого транспорта с коммуникацией пакетов. В общем случае этот транспорт, например, протокол UDP, не обеспечивает гарантированной доставки пакетов. Поэтому протокол L2TP самостоятельно решает эти вопросы, используя процедуры установления соединения внутри туннеля для каждого удаленного пользователя.

Следует отметить, что протокол L2TP не определяет конкретных методов криптозащиты и предполагает возможность применения различных стандартов шифрования. Если защищенный туннель планируется сформировать в IP-сетях, тогда для реализации криптозащиты используется протокол IPSec. Протокол L2TP поверх IPSec обеспечивает более высокую степень защиты данных, чем PPTP, так как использует алгоритмы шифрования 3-DES (Triple Data Encryption Standard) и AES. Кроме того, при помощи алгоритма HMAC (Hash Message Authentication Code) протокол L2TP обеспечивает аутентификацию данных. Для аутентификации данных этот алгоритм создает хэш длиной 128 разрядов.

Таким образом, функциональные возможности протоколов PPTP и L2TP различны. Протокол PPTP может применяться только в IP-сетях, и ему необходимо отдельное соединение TCP для создания и использования туннеля. Протокол L2TP может использоваться не только в IP-сетях, служебные сообщения для создания туннеля и пересылки по нему данных используют одинаковый формат и протоколы. Протокол L2TP поверх IPSec предлагает больше уровней безопасности, чем PPTP, и может гарантировать почти стопроцентную безопасность важных для организации данных.

Положительные качества протокола L2TP делают его весьма перспективным для построения виртуальных защищенных сетей. Однако при всех своих достоинствах протокол L2TP не способен преодолеть ряд недостатков туннельной передачи данных на канальном уровне:

* + - для реализации протокола L2TP необходима поддержка провайдеров ISP;
		- протокол L2TP ограничивает трафик рамками выбранного туннеля и лишает пользователей доступа к другим частям Интернета;
		- в протоколе L2TP не предусмотрено создания для текущей версии протокола IP криптозащищенного туннеля между конечными точками информационного взаимодействия;
		- предложенная спецификация L2TP обеспечивает стандартное шифрование только в IP-сетях с помощью протокола IPSec.

# Протоколы формирования защищенных каналов на сеансовом уровне

Самым высоким уровнем модели OSI, на котором возможно формирование защищенных виртуальных каналов, является пятый – сеансовый – уровень. При построении защищенных виртуальных сетей на сеансовом уровне появляется возможность криптографической защиты информационного обмена, включая аутентификацию, а также реализацию ряда функций посредничества между взаимодействующими сторонами.

Действительно, сеансовый уровень модели OSI отвечает за установку логических соединений и управление этими соединениями. Поэтому существует возможность применения на этом уровне программ-посредников, проверяющих допустимость запрошенных соединений и обеспечивающих выполнение других функций защиты межсетевого взаимодействия.

Протоколы формирования защищенных виртуальных каналов на сеансовом уровне прозрачны для прикладных протоколов защиты, а также высокоуровневых протоколов предоставления различных сервисов (протоколов HTTP, FTP, POP3, SMTP и др.). Однако на сеансовом уровне начинается непосредственная зависимость от приложений, реализующих высокоуровневые протоколы. Поэтому реализация протоколов защиты информационного обмена, соответствующих этому уровню, в большинстве случаев требует внесения изменений в высокоуровневые сетевые приложения.

## 3.1. Протоколы SSL и TLS

Протокол SSL (Secure Socket Layer – протокол защищенных сокетов) был разработан компанией Netscape Communications совместно с RSA Data Security для реализации защищенного обмена информацией в клиент/серверных приложениях. В настоящее время протокол SSL применяется в качестве протокола защищенного канала, работающего на сеансовом уровне модели OSI.

Протокол SSL использует криптографические методы защиты информации для обеспечения безопасности информационного обмена. Этот протокол выполняет все функции по созданию защищенного канала между двумя абонентами сети, включая их взаимную аутентификацию, обеспечение конфиденциальности, целостности и аутентичности передаваемых данных. Ядром протокола SSL является технология комплексного использования асимметричных и симметричных криптосистем.

Взаимная аутентификация обеих сторон в SSL выполняется путем обмена цифровыми сертификатами открытых ключей пользователей (клиента и сервера), заверенными цифровой подписью специальных сертификационных центров. Протокол SSL поддерживает сертификаты, соответствующие общепринятому стандарту Х.509, а также стандарты инфраструктуры открытых ключей PKI (Public Key Infrastructure), с помощью которой организуются выдача и проверка подлинности сертификатов.

Конфиденциальность обеспечивается шифрованием передаваемых сообщений с использованием симметричных сессионных ключей, которыми стороны обмениваются при установлении соединения. Сессионные ключи передаются также в зашифрованном виде, при этом они шифруются с помощью открытых ключей, извлеченных из сертификатов абонентов. Использование для защиты сообщений симметричных ключей связано с тем, что скорость процессов шифрования и расшифрования на основе симметричного ключа существенно выше, чем при использовании несимметричных ключей.

Подлинность и целостность циркулирующей информации обеспечивается за счет формирования и проверки электронной цифровой подписи. Для цифровых подписей и обмена ключами шифрования используются алгоритмы с открытым ключом.

В качестве алгоритма асимметричного шифрования используются алгоритм RSA. Допустимыми алгоритмами симметричного шифрования являются RC2, RC4, DES, 3-DES и AES. Для вычисления хэш-функций могут применяться стандарты MD5 и SHA-1. В протоколе SSL версии 3.0 набор криптографических алгоритмов является расширяемым.

Согласно протоколу SSL, криптозащищенные туннели создаются между конечными точками виртуальной сети. Инициаторами каждого защищенного туннеля являются клиент и сервер, функционирующие на компьютерах в конечных точках туннеля (рис. 10).

Протокол SSL предусматривает следующие этапы взаимодействия клиента и сервера при формировании и поддержке защищаемого соединения:

* + - установление SSL-сессии;
		- защищенное взаимодействие.



***Рис. 10. Криптозащищенные туннели, сформированные на основе протокола SSL***

В процессе установления SSL-сессии решаются следующие задачи:

* + - аутентификация сторон;
		- согласование криптографических алгоритмов и алгоритмов сжатия, которые будут использоваться при защищенном информационном обмене;
		- формирование общего секретного мастер-ключа;
		- генерация на основе сформированного мастер-ключа общих секретных сеансовых ключей для криптозащиты информационного обмена.

Процедура установления SSL-сессии, называемая также процедурой рукопожатия, отрабатывается перед непосредственной защитой информационного обмена и выполняется по протоколу начального приветствия (Handshake Protocol), входящему в состав протокола SSL. При установлении повторных соединений между клиентом и сервером стороны могут, по взаимному соглашению, формировать новые сеансовые ключи на основе старого общего секрета (данная процедура называется продолжением SSL-сессии).

В реализациях протокола SSL для аутентификации взаимодействующих сторон и формирования общих секретных ключей чаще всего используют алгоритм RSA.

Соответствие между открытыми ключами и их владельцами устанавливается с помощью цифровых сертификатов, выдаваемых специальными центрами сертификации .

В протоколе SSL предусмотрены два типа аутентификации:

* + - аутентификация сервера клиентом;
		- аутентификация клиента сервером.

SSL-аутентификация сервера позволяет клиенту проверить подлинность сервера. Клиентское ПО, поддерживающее SSL, может с помощью стандартных приемов криптографии с открытым ключом проверить, что сертификат сервера и открытый ключ действительны и были выданы источником, находящимся в списке доверенных источников сертификатов этого клиента. Это подтверждение может быть важным, если пользователь, например, отправляет номер кредитной карты по сети и хочет проверить подлинность сервера-получателя.

SSL-аутентификация клиента позволяет серверу проверить личность пользователя. Используя те же приемы, что и в случае с аутентификацией сервера, серверное ПО с поддержкой SSL может проверить, что сертификат клиента и открытый ключ действительны и были выданы источником сертификатов, имеющимся в списке доверенных источников сервера. Это подтверждение может быть важным, если, например, сервер – это банк, отправляющий конфиденциальную финансовую информацию заказчику, и он хочет проверить личность получателя. Процесс аутентификации клиента сервером иллюстрируется рис. 11.

***Рис. 11. Процесс аутентификации клиента сервером***

Протокол SSL прошел проверку временем, работая в популярных браузерах Internet Explorer и Netscape Navigator, а также на вебсерверах ведущих производителей.

В 1999 году появился протокол TLS (Transport Layer Security), который базируется на протоколе SSL 3.0 и в настоящее время является стандартом Интернета. Различия между протоколами SSL 3.0 и TLS не слишком существенны.

Спецификации SSL были в свое время предложены в качестве официальных стандартов Интернета, но не получили этого статуса по формальным обстоятельствам. Протокол SSL стал промышленным протоколом, развиваемым и продвигаемым вне технических координирующих институтов Интернета.

Некоторые *функции безопасности*, предоставляемые протоколом SSL:

* + - шифрование данных с целью предотвратить раскрытие конфиденциальных данных во время передачи;
		- подписывание данных с целью предотвратить несанкционированное изменение данных во время передачи;
		- аутентификация клиента и сервера, позволяющая убедиться, что общение ведется с соответствующим человеком или компьютером.

Протокол SSL поддерживается программным обеспечением серверов и клиентов, выпускаемых ведущими западными компаниями. Существенным недостатком протокола SSL является то, что практически все продукты, поддерживающие SSL, из-за экспортных ограничений доступны за пределами США лишь в усеченном варианте. Следует отметить, что последние экспортные релизы этих продуктов все же поддерживают ряд алгоритмов с достаточной длиной ключа, но с особыми ограничениями. Возникают также трудности создания и использования национальных центров сертификации.

К недостаткам протоколов SSL и TLS можно отнести то, что для транспортировки своих сообщений они используют только один протокол сетевого уровня – IP – и, следовательно, могут работать лишь в IP-сетях.

Как и другие программные продукты, SSL подвержен атакам, связанным с недоверенной программной средой, внедрением программ-закладок.

# Защита беспроводных сетей

Беспроводные сети начинают использоваться практически во всем мире. Это обусловлено их удобством, гибкостью и сравнительно невысокой стоимостью. Беспроводные технологии должны удовлетворять ряду требований к качеству, скорости, радиусу приема и защищенности, причем защищенность часто является самым важным фактором.

Сложность обеспечения безопасности беспроводной сети очевидна. Если в проводных сетях злоумышленник должен сначала получить физический доступ к кабельной системе или оконечным устройствам, то в беспроводных сетях это условие отпадает само собой: поскольку данные передаются «по воздуху», для получения доступа достаточно обычного приемника, установленного в радиусе действия сети.

Однако, несмотря на различия в реализации, подход к безопасности беспроводных сетей и их проводных аналогов идентичен: здесь присутствуют такие же требования к обеспечению конфиденциальности и целостности передаваемых данных и, конечно же, к проверке подлинности как беспроводных клиентов, так и точек доступа.

***Общие сведения***

Как и все стандарты IEEE 802, стандарт 802.11 работает на нижних двух уровнях модели ISO/OSI – физическом и канальном. Любое сетевое приложение, сетевая операционная система или протокол (например, TCP/IP) будут так же хорошо работать в сети 802.11, как и в сети Ethernet.

Стандарт 802.11 определяет два режима работы беспроводной сети – режим клиент/сервер (или режим инфраструктуры) и режим точка–точка (Ad-hoc).

В режиме клиент/сервер беспроводная сеть состоит как минимум из одной точки доступа AP (Access Point), подключенной к проводной сети, и некоторого набора беспроводных оконечных станций. Такая конфигурация носит название базового набора служб BSS (Basic Service Set). Два или более BSS, образующих единую подсеть, формируют расширенный набор служб ESS (Extended Service Set). Так как большинству беспроводных станций требуется получать доступ к файловым серверам, принтерам, Интернету, доступным в проводной локальной сети, они будут работать в режиме клиент/сервер.

Режим точка–точка (Ad-hoc) – это простая сеть, в которой связь между многочисленными станциями устанавливается напрямую, без использования специальной точки доступа. Такой режим полезен в том случае, если инфраструктура беспроводной сети не сформирована (например, отель, выставочный зал, аэропорт).

На физическом уровне стандарта 802.11 определены два широкополосных радиочастотных метода передачи и один – в инфракрасном диапазоне. Радиочастотные методы работают в ISM-диапазоне 2,4 ГГц и обычно используют полосу 83 МГц от 2,400 до 2,483 ГГц. Технологии широкополосного сигнала, используемые в радиочастотных методах, увеличивают надежность, пропускную способность, позволяют многим не связанным друг с другом устройствам разделять одну полосу частот с минимальными помехами друг для друга.

Основное дополнение, внесенное 802.11b в основной стандарт, – это поддержка двух новых скоростей передачи данных – 5,5 и 11 Мбит/с. Для достижения этих скоростей был выбран метод прямой последовательности DSSS (Direct Sequence Spread Spectrum). Канальный (Data Link) уровень 802.11 состоит из двух подуровней: управления логической связью LLC (Logical Link Control) и управления доступом к носителю MAC (Media Access Control).

***Обеспечение безопасности беспроводных сетей***

Система защиты беспроводных сетей WLAN, основанная на протоколе WEP (Wired Equivalent Privacy), первоначального стандарта 802.11 страдает существенными недостатками. К счастью, появились более эффективные технологии обеспечения информационной безопасности WLAN, которые описаны в стандарте WPA (WiFi Protected Access) организации Wi-Fi Alliance и стандарте 802.11i института IEEE и призваны устранить недостатки стандарта 802.11. Поскольку процесс разработки стандарта 802.11i слишком затянулся, организация Wi-Fi Alliance была вынуждена предложить в 2002 году собственную технологию обеспечения информационной безопасности WLAN – стандарт WPA.

Стандарт WPA весьма привлекателен тем, что относительно прост в реализации и позволяет защитить ныне действующие WLAN. Стандарты WPA и 802.11i совместимы друг с другом, поэтому использование поддерживающих WPA продуктов можно считать начальным этапом перехода к системе защиты на базе стандарта 802.11i.

Между технологиями 802.11i и WPA много общего. Так, в них определена идентичная архитектура системы безопасности с улучшенными механизмами аутентификации пользователей и протоколами распространения и обновления ключей. Но есть и существенные различия. Например, технология WPA базируется на протоколе динамических ключей TKIP (Temporal Key Integrity Protocol), поддержку которого в большинстве устройств WLAN можно реализовать путем обновления их ПО, а в более функциональной концепции 802.11i предусмотрено использование нового стандарта шифрования AES (Advanced Encryption Standard), с которым совместимо лишь новейшее оборудование для WLAN.

В стандарте WPA предусмотрено использование защитных протоколов 802.1x, EAP, TKIP и RADIUS.

Механизм аутентификации пользователей основан на протоколе контроля доступа 802.1x (разработан для проводных сетей) и протоколе расширенной аутентификации EAP (Extensible Authentication Protocol). Последний позволяет сетевому администратору задействовать алгоритмы аутентификации пользователей посредством сервера RADIUS.

Функции обеспечения конфиденциальности и целостности данных базируются на протоколе TKIP, который, в отличие от протокола WEP, использует более эффективный механизм управления ключами, но тот же самый алгоритм RC4 для шифрования данных.

Согласно протоколу TKIP, сетевые устройства работают с 48-битным вектором инициализации (в отличие от 24-битного вектора инициализации протокола WEP) и реализуют правила изменения последовательности его битов, что исключает повторное использование ключей и осуществление повторных атак.

В протоколе TKIP предусмотрены генерация нового ключа для каждого передаваемого пакета и улучшенный контроль целостности сообщений с помощью криптографической контрольной суммы MIC (Message Integrity Code), препятствующей хакеру изменять содержимое передаваемых пакетов.

Система сетевой безопасности стандарта WPA работает в двух режимах: PSK (Pre-Shared Key) и Enterprise (корпоративный). Для развертывания системы, работающей в режиме PSK, необходим разделяемый пароль. Такую систему несложно устанавливать, но она защищает WLAN не столь надежно, как это делает система, функционирующая в режиме Enterprise с иерархией динамических ключей. Хотя протокол TKIP работает с тем же самым блочным шифром RC4, который предусмотрен спецификацией протокола WEP, технология WPA защищает данные надежнее последнего.

Чтобы точки доступа WLAN стали совместимыми со стандартом WPA, достаточно модернизировать их ПО. Для перевода же сетевой инфраструктуры на стандарт 802.11i потребуется новое оборудование, поддерживающее алгоритм шифрования AES. Дело в том, что AES-шифрование создает большую нагрузку на центральный процессор беспроводного клиентского устройства.

Чтобы корпоративные точки доступа работали в системе сетевой безопасности стандарта WPA или 802.11i, они должны поддерживать аутентификацию пользователей по протоколу RADIUS и реализовывать предусмотренный стандартом метод шифрования – TKIP или AES, – что потребует модернизации их ПО. И еще одно требование – быстро осуществлять повторную аутентификацию пользователей после разрыва соединения с сетью. Это особенно важно для нормального функционирования приложений, работающих в реальном масштабе времени.

Если сервер RADIUS, применяемый для контроля доступа пользователей проводной сети, поддерживает нужные методы аутентификации EAP, то его можно задействовать и для аутентификации пользователей WLAN. В противном случае следует установить сервер WLAN RADIUS. Этот сервер работает следующим образом: сначала он проверяет аутентифицирующую информацию пользователя (на соответствие содержимому своей базы данных об их идентификаторах и паролях) или его цифровой сертификат, а затем активирует динамическую генерацию ключей шифрования точкой доступа и клиентской системой для каждого сеанса связи.

Для работы технологии WPA требуется механизм EAP-TLS (Transport Layer Security), тогда как в стандарте IEEE 802.11i применение конкретных методов аутентификации EAP не оговаривается. Выбор метода аутентификации EAP определяется спецификой работы клиентских приложений и архитектурой сети. Чтобы ноутбуки и карманные ПК работали в системе сетевой безопасности стандарта WPA или 802.11i, они должны быть оснащены клиентскими программами, поддерживающими стандарт 802.1x.

Самым простым, с точки зрения развертывания, вариантом системы сетевой безопасности стандарта WPA является система, работающая в режиме PSK. Она предназначена для небольших и домашних офисов и не нуждается в сервере RADIUS, а для шифрования пакетов и расчета криптографической контрольной суммы MIC в ней используется пароль PSK. Обеспечиваемый ею уровень информационной безопасности сети вполне достаточен для большинства вышеуказанных офисов. С целью повышения эффективности защиты данных следует применять пароли, содержащие не менее 20 символов.

Предприятиям целесообразно внедрять у себя системы сетевой безопасности стандарта WPA с серверами RADIUS. Большинство компаний предпочитают именно такие системы, поскольку работающие в режиме PSK решения сложнее администрировать, и они более уязвимы для хакерских атак.

До тех пор, пока средства стандарта 802.11i не станут доступными на рынке, WPA будет оставаться самым подходящим стандартом для защиты WLAN.

Стандарты WPA и 802.11i в достаточной степени надежны и обеспечивают высокий уровень защищенности беспроводных сетей. Тем не менее одного протокола защиты недостаточно – следует также уделить внимание правильному построению и настройке сети.

*Физическая защита.* При развертывании Wi-Fi-сети необходимо физически ограничить доступ к беспроводным точкам.

*Правильная настройка.* Парадокс современных беспроводных сетей заключается в том, что пользователи не всегда включают и используют встроенные механизмы аутентификации и шифрования.

*Защита пользовательских устройств.* Не следует полностью полагаться на встроенные механизмы защиты сети. Наиболее оптимальным является метод эшелонированной обороны, первой линией которой станут средства защиты, установленные на стационарном ПК или ноутбуке.

*Традиционные меры.* Эффективная работа компьютера в сети немыслима без классических мер защиты. Имеется в виду своевременная установка обновлений, использование защитных механизмов, встроенных в операционную систему и приложения, а также антивирусов. Однако этих мер на сегодня недостаточно, так как они ориентированы на защиту от уже известных угроз.

*Мониторинг сети.* Слабое звено в корпоративной сети – самовольно установленные точки доступа. Актуальной является задача локализации несанкционированных точек доступа. Специальные средства локализации точек доступа позволяют графически отображать место расположения «чужого» терминала на карте этажа или здания. Если классические методы не спасают от вторжения, на помощь приходят системы обнаружения атак.

*VPN-агенты.* Многие точки доступа работают в открытом режиме, поэтому необходимо использовать методы сокрытия передаваемых данных. На защищаемом компьютере должен быть установлен VPN-клиент, который возьмет на себя решение этой задачи. Практически все современные ОС содержат в своем составе такие программные компоненты.