АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Кафедра
Системы информационной безопасности

БЕЗОПАСНОСТЬ И АДМИНИСТРИРОВАНИЯ СИСТЕМ И СЕТЕЙ

Конспект лекции для студентов
5В100200 – Системы информационной безопасности

Алматы 2019 г.

СОСТАВИТЕЛЬ: К.М. Ургенишбаев. Безопасность и администрирования систем и сетей. Конспект лекции для студентов специальности 5В100200 – Системы информационной безопасности. – Алматы АУЭС, 2019. – 69 с

Конспект лекции содержит краткие теоретические сведения по дисциплине «Безопасность и администрирования систем и сетей» для 4-го курса, приводится список рекомендуемой литературы.

Ил.31, табл. 4, библиограф. – 7 назв.

Рецензент:

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2019 г.

© НАО «Алматинский университет энергетики и связи» на 2019 г.

Содержание

Лекция 1. I. Атаки 3
1. Обобщенный сценарий атаки 4
2. Классификация атак 6
3. Модель сетевой безопасности 9
Лекция 2. Криптография и системы шифрования 10
1. Стандартное шифрование 10
2. Типы атак на зашифрованные сообщения 12
3. Структура шифрования Фейстеля 14
4. Стандарт DES 15
Лекция 3. Методы аутентификации сообщений 19
1. Аутентификация с использованием стандартного шифрования 19
2. Односторонняя кэш функция 22
3. Безопасные кэш функции и HMAC 22
Лекция 4. Механизмы обеспечения безопасности коммутируемых локальных сетей 24
1. Ограничение количества управляющих компьютеров 24
2. Сегментация трафика (Traffic Segmentation) 26
3. Протокол IEEE 802.1 x 27
Лекция 5. Виртуальные сети (Virtual LAN) 32
1. Виртуальная ЛВС 32
2. Сети на базе портов (Port-based VLANs) 34
3. Построение виртуальных сетей 802.1 Q на одном коммутаторе. 36
Лекция 6. Механизмы обеспечения безопасности беспроводных локальных сетей 39
1. Классификация механизмов безопасности в сетях Wi-Fi 39
2. Блочное шифрование 41
3. Стандарт сети 802.11 i с повышенной безопасностью (WPA2) 45
Лекция 7. Виртуальные частные сети (Virtual Private Network, VPN) 48
1. Виртуальные частные сети 49
2. Обзор протоколов PPTP и L2TP 50
3. Протокол IPSEC 52
4. Ассоциация обеспечения безопасности. 53
5. Протокол аутентификации заголовка 55
6. Протокол SSL/TLS 58
Лекция 8. Безопасность удалённого управления 60
1. Аудит безопасности протокола SNMP 61
2. Модель протокола SNMP Общая модель 61
3. Протокол обмена данными 64
4. Формат SNMP-сообщения 65
4. Протокол SSH. 66

Введение

За несколько последних десятилетий требования к информационной безопасности существенно изменились. До начала широкого использования автоматизированных систем обработки данных безопасность информации достигалась исключительно физическими и административными мерами. С появлением компьютеров стала очевидной необходимость использования автоматических средств защиты файлов данных и программной среды. Следующий этап развития автоматических средств защиты связан с появлением распределенных систем обработки данных и компьютерных сетей, в которых средства сетевой безопасности, кроме защиты файлов и программной среды, необходимо использовать для защиты передаваемых по сетям данных. В наиболее полной трактовке под средствами сетевой безопасности мы будем иметь в виду меры предотвращения нарушений безопасности, которые возникают при передаче информации по сетям, а также меры, позволяющие определять, что такие нарушения безопасности имели место. Именно изучение средств сетевой безопасности и связанных с ними теоретических и прикладных проблем составляет основной материал лекции.

1 Лекция №1. Атаки

Вопросы лекции:

1) Обобщенный сценарий атаки.

2) Классы атак.

3) Модель сетевой безопасности

Цель: лекции ознакомить студентов с различными видами атак, осуществляемых хакерами на локальную сеть через интернет и способами защиты от этих атак.

Общее название совокупности средств, разработанных для защиты данных и предотвращения действий злоумышленников, есть компьютерная безопасность. Термин сетевая безопасность предполагает наличие средств для защиты данных при их передаче. Но на самом деле нет четких границ между этими двумя терминами.

Атака - любое действие, которое компрометирует безопасность информации, принадлежащей организации.

1.1 Обобщенный сценарий атаки

Рассмотрим обобщенный сценарий атаки, который можно представить в виде следующих шагов:

− пассивная разведка;

− активная разведка;

− выбор (разработка) эксплойта;

− взлом целевой системы;

− загрузка «полезного груза» (которым, как правило, является вредоносная программа);

− сокрытие следов взлома.

Конечно, данная последовательность может быть нарушена или могут быть исключены отдельные шаги данного сценария. Кратко рассмотрим этапы.

Пассивная разведка

Данный этап называется пассивным, так как злоумышленник не входит в непосредственный контакт с целью или входит с соблюдением общепринятых правил (например, посещая сайт компании, куда входит целевая система).

Целью данного этапа является сбор информации о цели. В качестве цели может выступать как конкретный хост (сервер), так и целая сеть организации. Достаточно часто цель выбирается из условия простого поиска системы, содержащей известную уязвимость, к которой злоумышленник имеет эксплойт.

Активная разведка

Этап проведения активной разведки называют сканированием. Злоумышленник пытается определить структуру интересующей его сети,

точки доступа, доступные узлы и хосты, расположение маршрутизаторов и межсетевых экранов, установленные операционные системы и их версии, открытые порты и работающие службы, версии установленных программных продуктов. На этом этапе злоумышленник входит в контакт с объектами интересующей его системы, поэтому его действия могут быть обнаружены средствами защиты целевой системы.

Для решения задач проведения активной разведки можно использовать большое количество разнообразных средств, многие из которых являются общедоступными (ping, traceroute, nmap, SuperScan).

Может использоваться «замедленное» сканирование, когда злоумышленник посылает отдельные запросы через большие интервалы времени. В этом случае достаточно высока вероятность того, что на целевой системе не обратят внимания на отдельные запросы.

Выбор эксплойта.

На основании полученных данных злоумышленник выбирает, модифицирует имеющийся или разрабатывает новый эксплойт для проведения взлома.

Взлом целевой системы.

Существует множество различных способов взлома. К их числу можно отнести получение доступа к системе, расширение имеющихся полномочий и отказ в обслуживании. Нападающий и защищающийся находятся в неравных условиях. Для защиты системы необходимо знание обо всех возможных атаках против данной системы, а для проведения атаки достаточно найти только одну уязвимость.

Загрузка «полезного груза»

Как правило, атака проводится не только ради самого процесса взлома. Обычно злоумышленник хочет иметь возможность возврата во взломанную систему или использования её в качестве плацдарма для атаки других систем. Кроме того, его может интересовать получение конфиденциальных данных во взломанной системе. К действию по загрузке «полезного кода» можно отнести:

− установка снифферов на взломанную систему для получения информации, позволяющей осуществить взлом соседей;

− осуществление перенаправления портов, чтобы обеспечить передачу пакетов со взломанной системы на другой компьютер;

− установка «патчей» на использованную уязвимость или имеющиеся уязвимости, чтобы исключить взлом системы другими злоумышленниками;

− создание фальшивых учётных записей, наделенных привилегиями суперпользователя;

− создание или установка готового потайного входа во взломанную систему;

− установка «троянских коней» для сбора конфиденциальной информации, последующего входа в систему или для проведения атак других систем.

Эти и другие операции проводятся при отключении аудита на взломанной системе.

Сокрытие следов взлома

На данном этапе используются программы удаления записей из различных журналов, ведущихся в системе, скрытие установленных файлов, использование потоков файлов в файловой системе NTFS, троянизирование системных программ и утилит и замена программных компонент операционной системы.

Для реализации этих целей используются специальные наборы средств (root kit). Первые наборы таких средств были разработаны в начале 90-х годов для ОС Unix. В настоящее время они существуют практически для всех операционных систем. Первые такие наборы представляли собой троянские файлы, в которые были встроены потайные ходы. Они предназначались для подмены наиболее часто используемых программ типа netstat.

Различают два основных вида таких наборов средств: наборы уровня ядра и наборы уровня пользователя. Это различие вызвано теми компонентами программного обеспечения, которые перезаписываются (изменяются) в атакованной системе. Чтобы использовать набор средств, атакующий должен сначала получить права администратора в атакованной системе.

Наибольшую опасность представляют наборы средств уровня ядра. Поскольку они имеют высшие права, то могут быть полностью скрыты от другого ПО, запущенного в системе. Как правило, в набор входят файлы, которые заменяют программные модули в ядре операционной системы или отдельные библиотеки, различные вспомогательные средства для получения необходимой информации, а также скрипты инсталляции. Одной из главных задач атакующего является обеспечение выживаемости установленных средств после перезагрузки и обеспечение их «невидимости» для пользователя или администратора.

1.2 Классы атак

В общем случае существует поток данных от источника информации к получателю. Нормальный поток отражен на рисунке 1.1а. Остальные части рисунка показывают четыре общие категории атак:

Прерывание: передаваемая информация уничтожается или становится недоступной. Этот тип атак направлен на доступность информации. Примерами могут служить уничтожение части аппаратного обеспечения, обрыв линии или блокировка файловой системы.

Перехват: неавторизованная сторона получает доступ к информации. Этот тип атак направлен на конфиденциальность информации. Неавторизованной стороной может быть человек или программа. Примерами могут служить перехват сообщений в сети и неавторизованное копирование файлов или программ.

Изменение: неавторизованная сторона не только получает доступ, но и фальсифицирует информацию. Этот тип атак направлен на целостность информации. Примерами могут служить изменение данных в файле, изменение кода программы и модифицирование содержимого сообщения, передаваемого по сети.

Фальсификация: неавторизованная сторона вставляет в поток информации поддельный объект. Этот тип атак направлен на подлинность информации. Примерами могут служить вставка случайных сообщений в сетевой поток информации или добавление записей в файл. Также атаки делятся на пассивные и активные (рисунок 1.2).

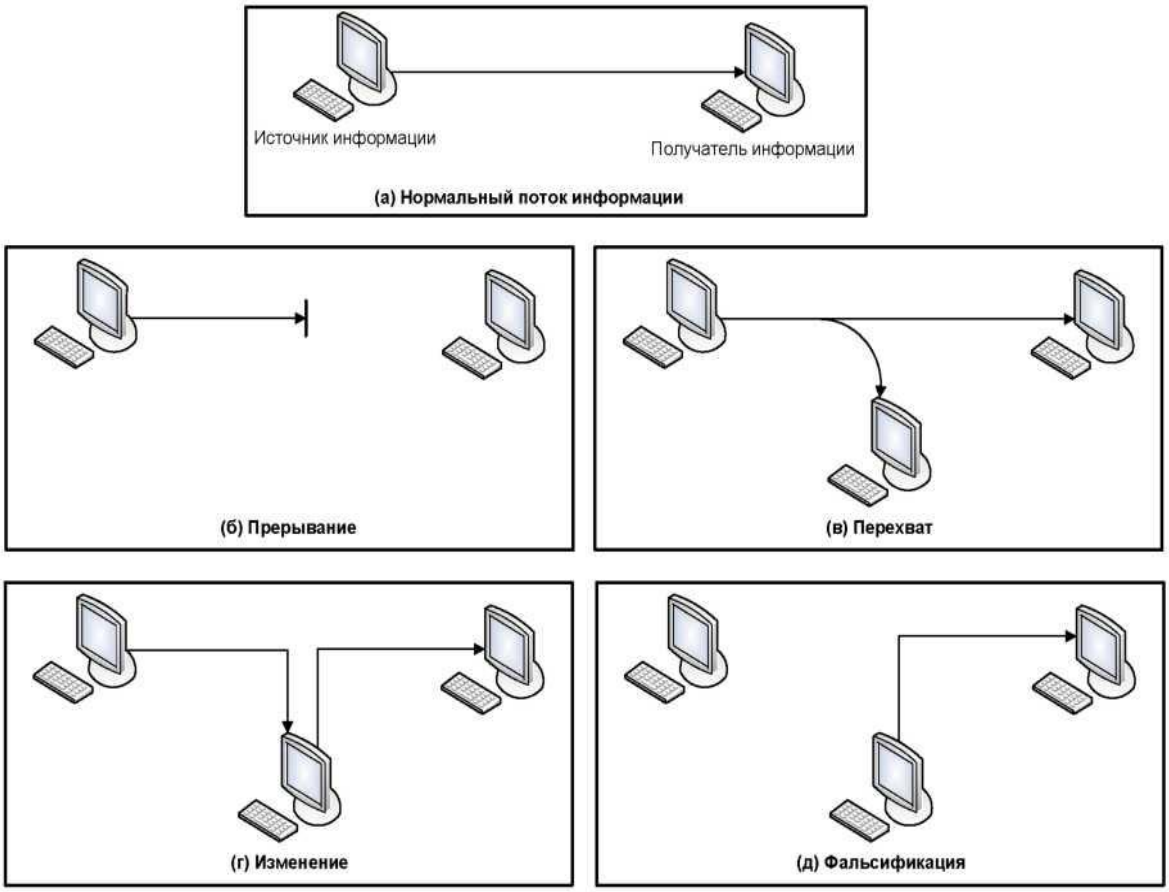


Рисунок 1.1 - Классификация атак по Столингсу

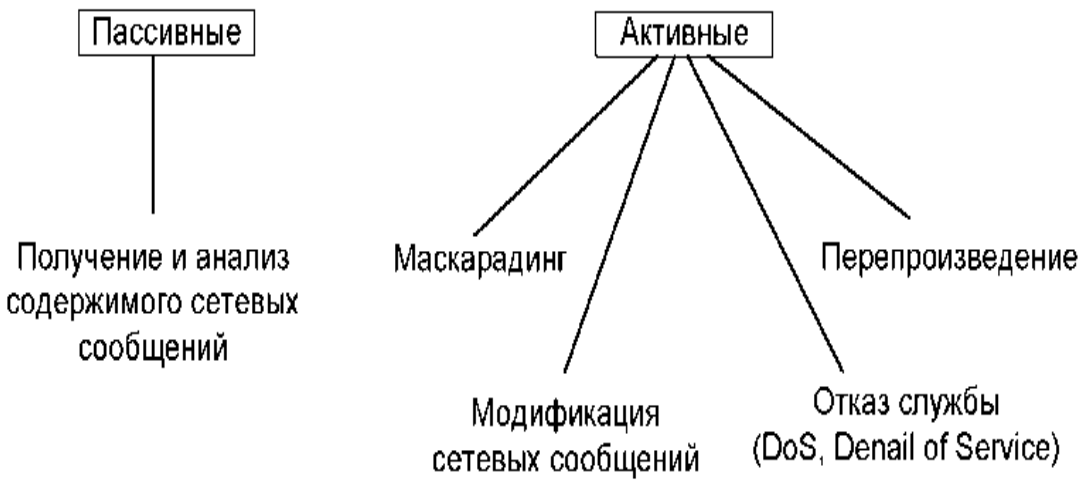


Рисунок1.2 – Пассивные и активные атаки

Пассивные атаки

Пассивные атаки по своей природе схожи с подслушиванием или слежкой. Целью оппонента является получение передаваемой информации. Существует два типа пассивных атак: получение содержимого сетевых сообщений и анализ сетевого трафика. Первый тип легок для понимания. Телефонная беседа, электронная почта, передаваемый файл и прочее могут содержать важную информацию.

Второй тип более тонок для понимания. Предположим, что мы можем маскировать содержимое сетевых сообщений таким образом, что даже если оппонент перехватит их, то все равно не сможет извлечь информацию из сообщения. Самый простой способ маскировки сообщений - это их шифрование. Тем не менее даже при наличии шифровки оппонент все равно в состоянии осуществить анализ перехваченного сообщения. Пассивные атаки очень трудны для обнаружения, так как они не изменяют содержимого сообщений. Тем не менее вполне реально предотвратить данный тип атак.

Активные атаки

Данные атаки производят модификацию потока данных или создают фальшивый поток. Делятся на четыре типа.

Маскарадинг имеет место когда одна сущность выдает себя за другую. Маскарадинг зачастую включает в себя какой-либо другой тип активных атак. Например, пакеты для прохождения процедуры аутентификации могут перехвачены и перепроизведены после того, как процедура аутентификации настоящего пользователя была успешно осуществлена. Таким образом, это даст злоумышленнику вход в систему под определенной учетной записью.

Перевоспроизведение включает пассивный перехват данных и их дальнейшую ретрансляцию. Модификация сообщений говорит само за себя.

Отказ службы (DoS) приводит к ненормальному функционированию какой-либо сетевой службы. Данный тип атак может также к неработоспособности всей сети в целом путем перегрузки ее сетевым «мусором».

1.3. Модель сетевой безопасности

Обобщенную модель сетевой безопасности можно представить в следующем виде (рисунок 1.3).

Грамотно построенная сеть (с точки зрения безопасности) должна содержать как минимум два периметра:

Внутренний. В данном периметре располагаются только компьютеры пользователей, которые входят в сеть незарегистрированных IP-адресов. Также данная сеть может содержать беспроводные точки доступа для подключения мобильных пользователей. Во внутреннем периметре применяются механизмы внутрисетевой безопасности.

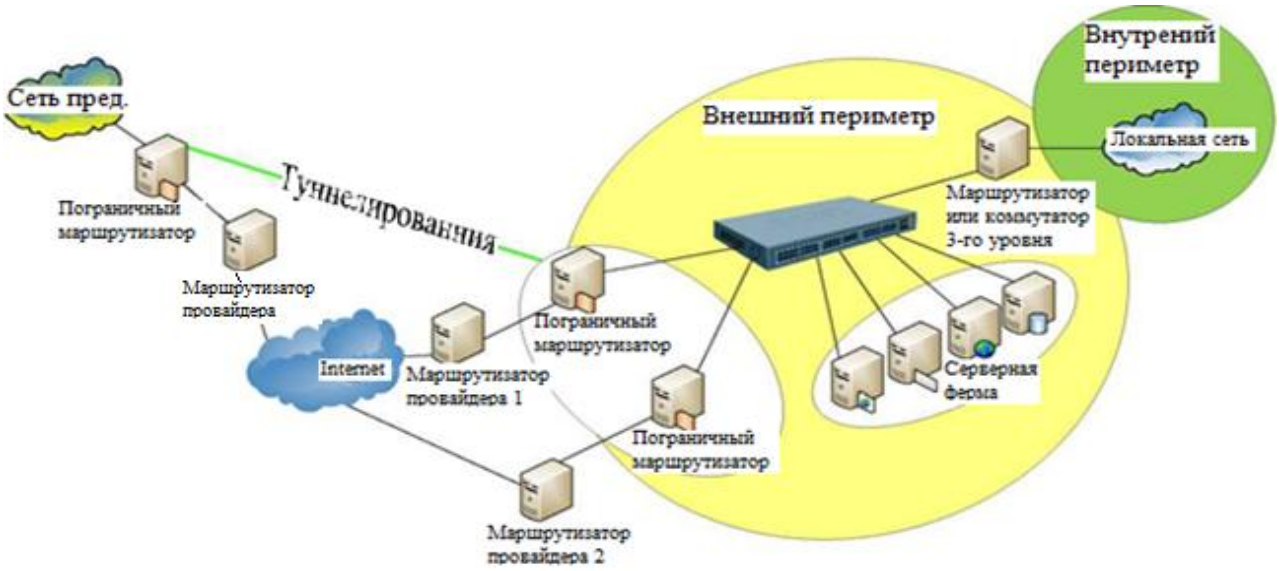


Рисунок 1.3 - Обобщенная архитектура сетевой безопасной сети.

Внешний периметр. Обычно содержит сервера предприятия (серверная ферма) и пограничные маршрутизаторы, обеспечивающие доступ во внешние сети. Все компьютеры во внешнем периметре обычно включаются в IP-сеть зарегистрированных адресов. В данном периметре применяются механизмы межсетевой безопасности, к которым относятся:

межсетевые экраны (МЭ);

системы трансляции адресов и портов;

системы обнаружения атак и вторжений (СОАВ);

системы туннелирования.

Между периметрами устанавливается маршрутизатор или коммутатор L3, на котором также могут применяться технологии межсетевой безопасности. Все маршрутизаторы могут как программными, так и аппаратными.

Данная концепция является расширяемой. Если необходимо повысить

уровень безопасности сети, то можно увеличить количество внутренних периметров. Следующим этапом развития модели является добавление ещё одного внутреннего периметра. В этом случае во внешнем периметре устанавливаются сервера-ловушки, в первом внутреннем периметре - сервера предприятия и только во втором внутреннем периметре - компьютеры пользователей.

Контрольные вопросы.

1. Что такое активная разведка;
2. Что такое пассивная разведка;
3. Как классифицируются виды атак;
4. Для чего используется технология NAT
5. Для чего используется система туннелированния.

Лекция 2. Криптография и системы шифрования

Вопросы лекции:

1) Стандартное шифрования.
2) Типы атак на зашифрованные сообщения.
3) Структура шифрования Фейстеля
4) Стандарт DES

Цель занятия ознакомить студентов различными видами шифрования исходного текста, а также типами атак на зашифрованное сообщения осуществляемых злоумышленниками.

1. Стандартное шифрование

Стандартное шифрование (симметричное шифрование, шифрование с использованием одного ключа) было единственным используемым типом шифрования до изобретения шифрования на основе открытого ключа в конце 1970-х годов. До сих пор стандартное шифрование остается широко используемым.

Алгоритм работы системы стандартного шифрования состоит из пяти шагов (рисунок 2.1):

Открытый текст (в дальнейшем для простоты просто «текст»): это оригинальное сообщение или данные, которые являются входными для алгоритма;

Алгоритм шифрования: это алгоритм, который выполняет различные подстановки и изменения над исходным текстом;

Секретный ключ: также является входным элементом для алгоритма. Конкретные подстановки и изменения, выполняемые алгоритмом шифрования, зависят от ключа;

Шифр-текст: это зашифрованное сообщение, получаемое на выходе алгоритма. Его вид зависит от исходного текста и секретного ключа. Для одного и того же исходного сообщения два различных секретных ключа на выходе алгоритма дадут два различных шифр-текста;

Алгоритм дешифрования: это алгоритм шифрования, работающий в обратном порядке. В качестве входных данных он берет шифр-текст и такой же секретный ключ и производит исходный текст.

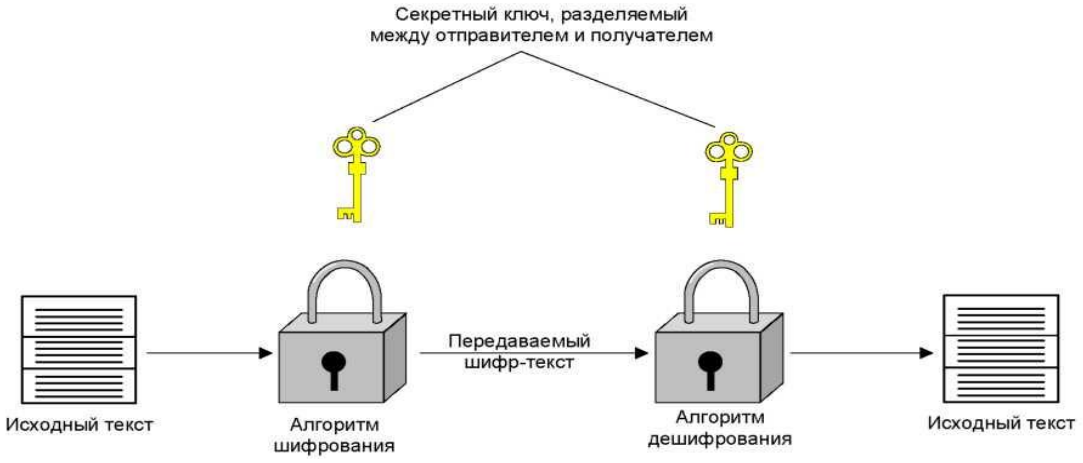


Рисунок 2.1 - Упрощенная модель работы алгоритма стандартного шифрования.

Существует два требования для безопасного использования стандартного шифрования:

Необходим мощный алгоритм шифрования. Как минимум необходимо, чтобы алгоритм был таким, что злоумышленник, знающий этот алгоритм и имеющий доступ к одному или более шифр-текстам, не смог бы их расшифровать или вычислить ключ. Данное требование обычно звучит в следующей, более строгой форме: злоумышленник не должен расшифровать шифр-текст или обнаружить ключ даже в том случае, если он владеет несколькими шифр-текстами вместе с исходными текстами, из которых они были получены.

Отправитель и получатель должны получать копии секретного ключа и хранить его безопасным образом. Если кто-либо узнает ключ и знает алгоритм, то все данные, шифруемые с использованием этого ключа, станут читабельны.

Важно заметить, что безопасность стандартного шифрования зависит от секретности ключа, а не от секретности самого алгоритма шифрования. То есть подразумевается, что задача расшифровки сообщения на базе шифр-текста и знания алгоритма шифрования/дешифрования является практически невозможной. Другими словами, нет необходимости хранить в секретности сам алгоритм; необходимо хранить в секретности только ключ.

Криптография

Криптографические системы в общем разделяются на три независимых класса:

Тип операций, используемых для преобразования исходного текста в шифр- текста. Все алгоритмы шифрования базируются на двух основных принципах: подстановка, при которой каждый элемент в исходном тексте

(бит, буква, группа бит или букв) представляются другим элементом, и перестановка, при которой элементы исходного текста меняются местами. Основное требование - не допустить потери информации, чтобы можно было в дальнейшем осуществить процесс расшифровки. Большинство систем включает не один, а несколько последовательных этапов замены и перестановки.

Количество используемых ключей. Если отправитель и получатель используют один и тот же ключ, система называется симметричной (с одним ключом, или с секретным ключом, или система стандартного шифрования). Если отправитель и получатель используют разные ключи, система называется ассиметричной (с двумя ключами или система шифрования на основе открытого ключа).

Способ обработки исходного текста. Блочный шифровщик обрабатывает на входе системы один блок элементов за раз, производя блок элементов на выходе для каждого входного блока. Поточный шифровщик обрабатывает входные элементы последовательно, производя на выходе один элемент за раз.

Криптоанализ

Процесс попытки раскрыть исходный текст или ключ называется криптоанализом. Стратегия, используемая шифроаналитиком, зависит от природы схемы шифрования и информации, доступной для шифроаналитика.

2. Типы атак на зашифрованные сообщения

Таблица 2.1 - Типы атак на зашифрованные сообщения

Тип атаки	Известно шифроаналитику
Только шифр-текст	Алгоритм шифрования Шифр-текст, который необходимо расшифровать
Известный исходный текст	Алгоритм шифрования Шифр-текст, который необходимо расшифровать. Один или более пар «исходный текст» - «шифр-текст»
Выбранный исходный текст	Алгоритм шифрования. Шифр-текст, который необходимо расшифровать. Текстовое сообщение, выбранное шифро-аналитиком совместно с соответствующим шифр-текстом.
Выбранный шифр-	Алгоритм шифрования
	Шифр-текст, который необходимо расшифровать Предполагаемый шифр-текст, выбранный шифро-аналитиком, совместно с соответствующим дешифрованным исходным текстом.
Выбранный текст	Алгоритм шифрования. Шифр-текст, который необходимо расшифровать. Сообщение исходного текста, выбранное шифроаналитиком, совместно с соответствующим шифр-текстом. Предполагаемый шифр-текст, выбранный шифроаналитиком, совместно с соответствующим дешифрованным исходным текстом

Таблица 2.1 показывает различные типы криптоаналитических атак, основанных на количестве информации, известной шифроаналитику.

Наиболее сложная проблема - это, когда вся известная информация представлена только шифр-текстом. Тем не менее в большинстве случаев алгоритм шифрования все же известен. Одной из возможных атак в этом случае может быть «Брут-форс» атака («грубая сила»), которая заключается в переборе всех возможных ключей. Если множество ключей слишком большое, подобная задача становится практически невозможной. Таким образом, злоумышленник должен полагаться на анализ самого шифр-текста, применяя к нему различные статистические тесты. Для использования этого метода злоумышленник должен предполагать тип данных, содержащихся в исходном тексте (например, английский или французский текст, исполняемый файл какой-либо ОС, java-скрипт и т.д.).

Атаке под названием «только шифр-текст» легче всего противостоять, так как злоумышленник обладает минимальным количеством информации для своей работы. Тем не менее в большинстве случаев шифроаналитик обладает большим количеством информации. Аналитик может перехватить одно или более сообщений с исходным текстом также, как и их шифровки. Или же аналитик может знать, что конкретный кусок исходного появится в сообщении. Например, файл, представляемый в формате Postscript, всегда начинается с одного и того же сочетания символов. Все вышеприведенные случаи относятся к типу атак под названием «известный исходный текст». Зная эту информации, аналитик может алгоритмически вывести значение секретного ключа.

Тесно связана с атакой «известный исходный текст» атака под названием «вероятное слово». Если злоумышленник работает с шифровкой какого-либо типичного сообщения, то он может догадываться о содержимом части этого сообщения. Примером такого сообщения может служить исходный код программы, разработанный корпорацией, и этот исходный код может содержать копирайт в какой-то стандартной позиции.

Если аналитик в состоянии получить доступ к системе шифрования, чтобы затем подсунуть ей какой-либо сообщение, то подобный тип атак называется атака «выбранный исходный текст». В общем случае, если аналитик в состоянии выбрать сообщение для шифровки, то тогда аналитик может выборочно подбирать образцы текста для последующего анализа их шифровок с целью раскрытия секретного ключа.

Таблица 2.1 содержит два других типа атак: «выбранный шифр-текст» и «выбранный текст». Он возможны чисто теоретически, но не практически. Атаки с использованием только шифр-текста не сможет противостоять только очень слабый алгоритм шифрования. В общем случае алгоритмы шифрования разрабатываются таким образом, чтобы противостоять как минимум атакам «известный исходный текст».

Необходимо рассмотреть еще два определения. Схема шифрования является безопасной с точки зрения вычислений, если шифр-текст, сгенерированный этой схемой отвечает одному или более условиям:

− стоимость взлома шифра превышает ценность зашифрованной

информации;

− время взлома превышает полезное время жизни информации.

Проблем заключается в том, что очень трудно вычислить количество попыток, требуемых для удачного взлома шифра. Тем не менее, если предположить, что алгоритм шифрования не имеет никаких математических изъянов и, следовательно, для его взлома применима только брут-форс атака, то тогда мы можем сделать некоторые предположения относительно времени и стоимости взлома.

3. Структура шифрования Фейстеля

Все алгоритмы стандартного блочного шифрования имеют структуру, впервые описанную сотрудником IBM Хорстом Фейстелем в 1973 году. Данная структура показана на рисунке

Входными данными для алгоритма шифрования являются блок текста длинной 2w бит и ключ K. Блок текста разбивается на две половины L0 и R0. Эти две половинки данных проходят обработку в течении n кругов (раундов), а затем объединяются для получения блока шифр-текста. Каждый i-ый раунд в качестве входных данных берет Li-1 и Ri-1, полученное с выхода предыдущего раунда, и ключ Ki, полученный из общего ключа K. В общем случае подключи Ki отличаются от K и друг от друга. Подключи Ki генерируются из ключа K алгоритмом генерации подключей.

Все раунды имеют одну и ту же структуру. Подстановка выполняется над левой половиной данных. Это процедура осуществляется следующим образом:

выполняется функция F над правой половиной данных;

выполняется побитовая операция исключающего ИЛИ (XOR) над выходными данными функции F и левой половиной данных.

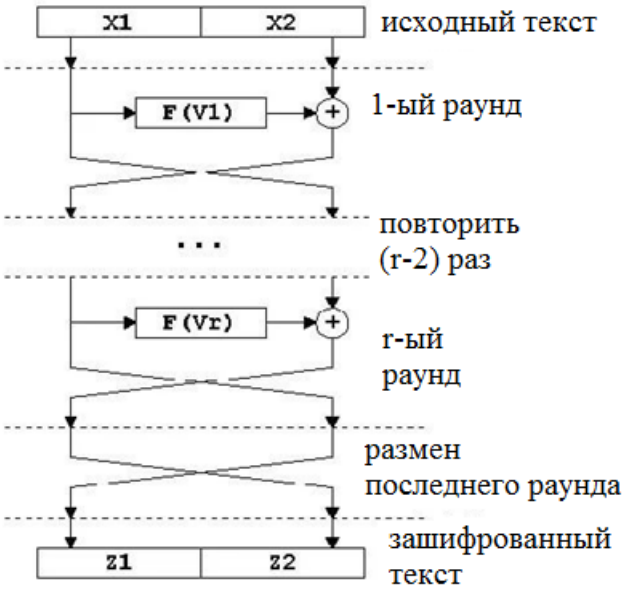


Рисунок 2.2 - Классическая сеть Фейстеля.

Функция F имеет одну и ту же структуру для всех раундов, но разные входные подключи K|. В конце каждого раунда происходит перестановка левой и правой половин данных. Конкретная реализация сети Фейстеля зависит от выбора следующих параметров:

− S размер блока данных. Чем больше блок, тем больше безопасность и ниже скорость обработки. Стандартным и рекомендуемым считается блок в 64 бита.

− размер ключа. Чем больше ключ, тем больше безопасность и ниже скорость обработки. Наиболее распространенный размер ключа в современных алгоритмах - 128 бит.

− количество раундов. Один раунд не обеспечивает должной безопасности. Чем больше раундов, тем выше безопасность. Стандартным является 16 раундов.

− функция F. Чем сложнее функция, тем сложнее алгоритм шифрования более устойчив ко взлому.

Есть также два других соображения по поводу архитектуры сети Фейстеля:

− скорость ПО шифрования/дешифрования. В большинстве случаев шифрование алгоритм шифрования включается в ПО, дабы исключить его аппаратную реализацию. Соответственно скорость выполнения алгоритма становится важным аспектом.

− легкость анализа. Несмотря на то, что нам хотелось бы сделать наш алгоритм как можно более сложным для криптоанализа, существует некоторые плюсы, если алгоритм легко анализировать. То есть, если алгоритм может быть легко и кратко объяснен, в нем легче отыскать уязвимости и тем самым повысить уровень его защищенности.

Процесс дешифрования на основе сети Фейстеля в основном схож с процессом шифрования. В качестве входных данных используются блок шифр-текста и подключи Ki, но в обратном порядке. То есть необходимо использовать Kn в первом раунде и K1 в последнем раунде. Это очень хорошо, так как нам не нужно использовать различные алгоритм для шифрования и дешифрования.

Алгоритмы стандартного шифрования

Наиболее распространенные алгоритмы стандартного шифрования являются блочными. Наиболее важными из них являются стандарт DES (Data Encryption Standard, Стандарт Шифрования Данных) и алгоритм TDEA (Triple Data Encryption Algorithm, Алгоритм Тройного Шифрования Данных).

4. Стандарт DES

Наиболее широко используемой схемой шифрования является стандарт DES, принятый в 1977 Национальным Бюро Стандартов США (сейчас Национальный Институт Стандартов и Технологий) в качестве Федерального Стандарта Обработки Информации. В 1994 году НИСТ утвердил DES для

федерального использования в течении следующих пяти лет. Алгоритм, лежащий в основе стандарта, называется DEA.

Описание алгоритма

Общая схема шифрования DES представлена на рисунке 2.3.

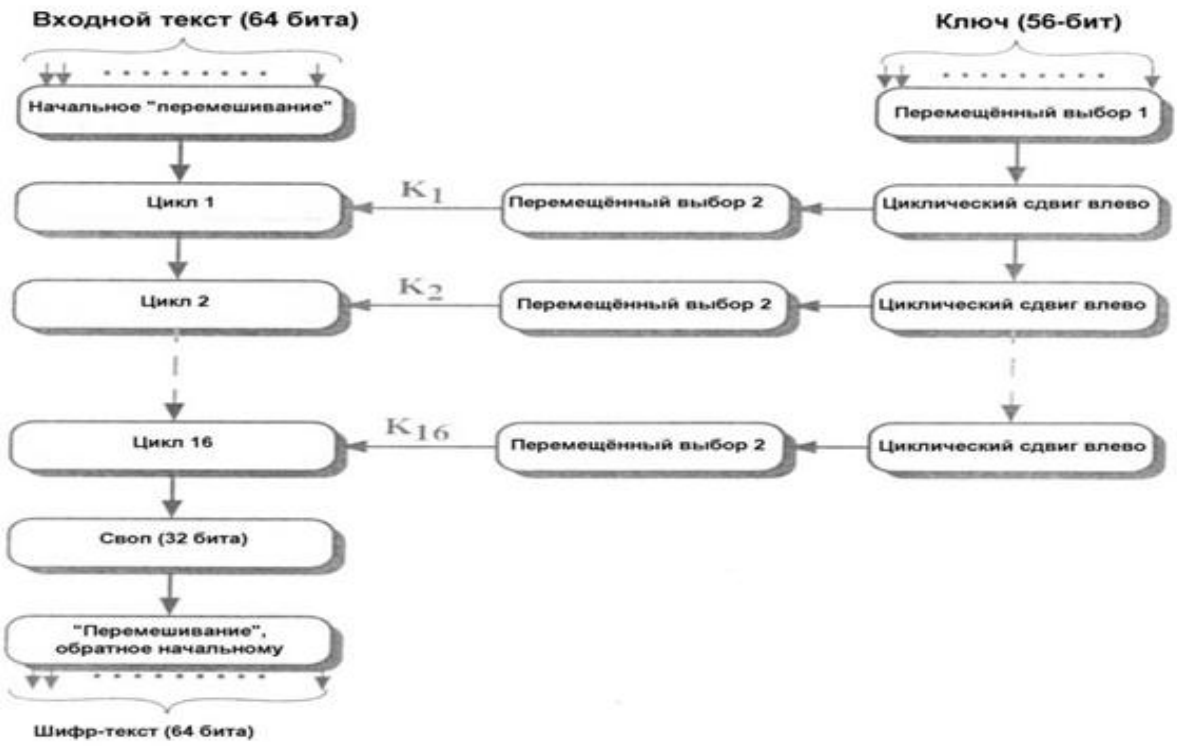


Рисунок 2.3 - Общее представление алгоритма шифрования DES.

Исходный текст разбивается на блоки в 64 бита, а ключ имеет размер 56 бит. Левая часть рисунка показывает, что обработка текста происходит в три фазы. В первую очередь, 64 битный текст проходит начальную перестановку (НП), которая меняет местами биты. Затем следует фаза, состоящая из 16 итераций одной и той же функции. На выходе последней итерации производится перестановка местами левой и правой половин 64-битного текста. В конце производится конечная перестановка (КП), которая является инверсной относительно функции НП и выдает 64-битный шифр-текст.

Правая часть рисунка 3.3 показывает способ использования 56-бтового ключа. Изначально ключ проходит через процедуру перестановки бит. Далее для каждой из 16 итераций генерируется подключ Ki путем циклического сдвига влево, а затем перестановки бит.

Функция перестановки одна и та же для каждой итерации, но при этом

генерируются различные подключи из-за того, что на каждом этапе производится циклический сдвиг влево.

Рисунок 3.4 более детально представляет алгоритм для единичной итерации. На входе каждой итерации 64-битовой текст разделяется на две 32-битовые половины, которые помечаются как L и R соответственно. Весь процесс для каждой итерации может быть выражен через следующие формулы:

Ц = Rn, Ri = LM 0 F(RM, Ki)

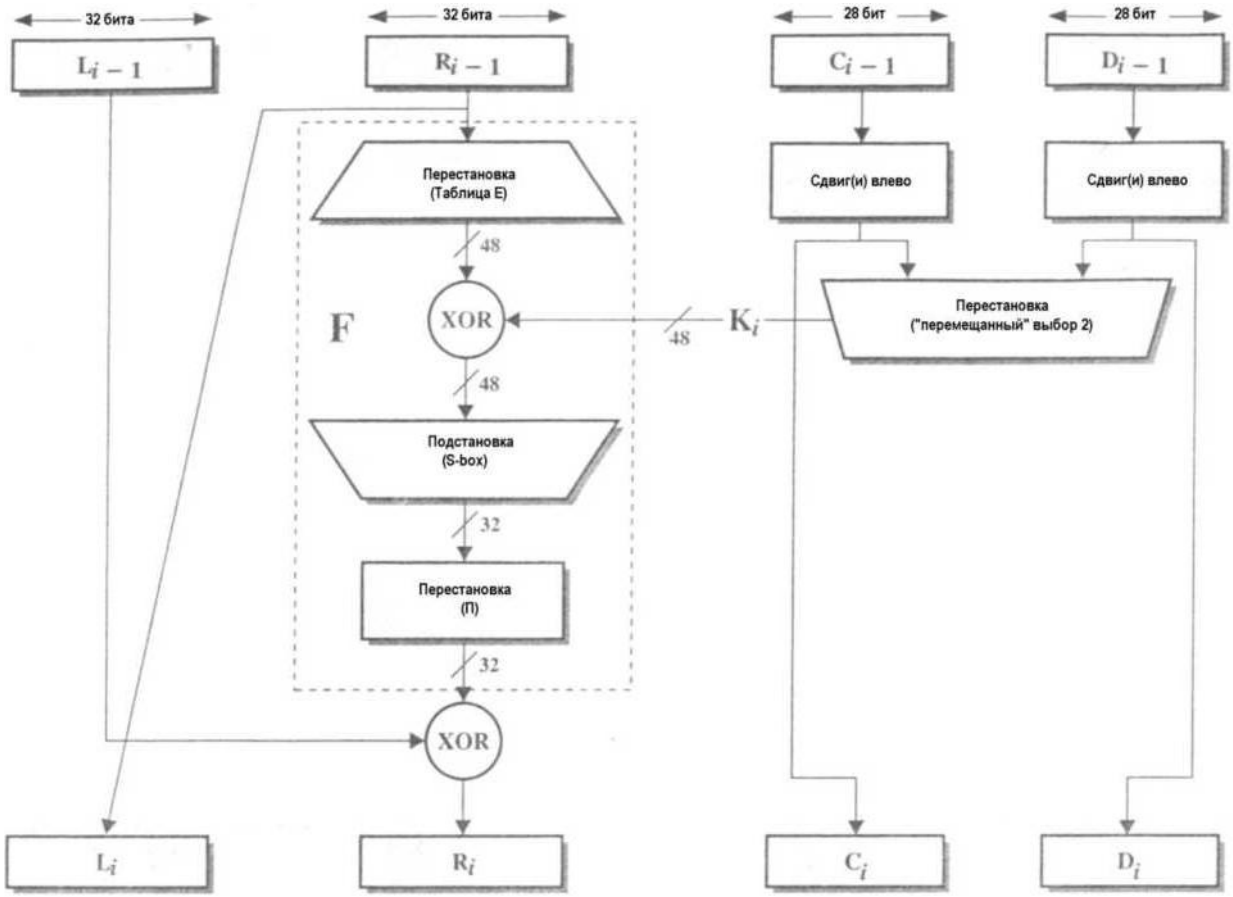


Рисунок 2.4 - Алгоритм одной итерации DES.

Таким образом, левая часть данных на выходе итерации является идентичной правой части данных на входе итерации. Правая часть данных на выходе итерации является результатом операции побитового исключающего ИЛИ, произведенной над Li-1 и результатом комплексной функции F. Данная функция включает как операции перестановки, так и подстановки. Операция подстановки, представленная на блок-схеме вершиной «S-box», просто отображает комбинацию из 48 входных бит в соответствующий паттерн из 32 бит.

Возвращаясь к рисунку 2.3, видно, что 56-битовый ключ сначала проходит процедуру перестановки. Полученный 56-битовый ключ далее разделяется на две половины по 28 бит каждая, которые маркируются как C0

и D0. На каждой итерации половины C и D отдельно подвергаются операции циклического сдвига влево на 1 или 2 бита. Полученные путем сдвига величины служат входными половинами для следующей итерации, а также для функции перестановки.

Процесс дешифровки DES в основном тот же самый, что и процесс шифровки. При этом действуют следующие правила: использовать шифр-текст в качестве входных данных, и ключи Ki в обратном порядке.

Устойчивость DES

Споры относительно устойчивости алгоритма DES делятся на две области: относительно собственно самого алгоритма и относительно использования 56-битового ключа. Многие годы осуществлялось большое количество попыток найти и использовать слабые стороны этого алгоритма, что сделало DES наиболее изученным алгоритмом. Несмотря на это никто так и не нашел слабых мест в алгоритме DES, во всяком случае не опубликовал. Более серьезным спором является проблемы длины ключа. Еще в начале 70-х годов эксперты предупреждали, что в плане безопасности дни алгоритма DES сочтены, это лишь вопрос времени роста скорости работы процессов и их стоимости. Это время наступило в июле 1998 года, когда фонд «Электронная Граница» объявил, что он взломал алгоритм DES, используя специальный компьютер «Взломщик DES», стоимость на создание которой оказались меньше $250000. Атака длилась менее трех дней. Фонд опубликовал детальное описание машины, тем самым предоставив другим людям возможность построить собственные взломщики. Цены на аппаратное обеспечение падают, а производительность процессоров растет, делая DES все более слабым.

Важно отметить, что для взлома была применена не чистая брут-форс атака - осуществлялся поиск нужного ключа, а не простой перебор всех возможных ключей. Для удачного взлома аналитик должен распознать исходный текст.

Если бы единственной формой атаки была брут-форс атака, тогда способ противостояния данной атаки очевиден - использование более длинного ключа. При этом необходимо подумать о приемлемой длине ключа. Для этого воспользуемся взломщиком Фонда Электронная Граница. Данный взломщик был прототипом и следует предполагать, что на сегодняшний день возможно создать более быструю машину. Предположим, что взломщик может перебирать один миллион ключей за одну микросекунду, тогда алгоритм DES будет взломан не более, чем за 10 часов. Это примерно в 7 раз быстрее, чем время, затраченное Фондом. Используя эту скорость для взлома DES-подобного алгоритма с 128битовым ключом понадобится примерно 1018 лет. Таким образом, 128-битовый ключ является гарантом, что алгоритм не будет взломан.

Контрольные вопросы.

1. Алгоритма шифрования DES;

2. Структура шифрования Фейстеля;
3. Где используется алгоритм симметричного шифрования;
4. Что такое КОК;
5. Что такое криптография.

Лекция 3. Методы аутентификации сообщений

Вопросы лекции:
1) Аутентификация с использованием стандартного шифрования.
2) Односторонняя кэш функция.
3) Безопасные кэш функции и HMAC.

Цель занятия ознакомить студентов различными видами шифрования исходного текста, а также типами атак на зашифрованное сообщения осуществляемых злоумышленниками.

Шифрование защищает от пассивных атак (прослушивание содержимого трафика сети). Совсем другие требования предъявляются к защите против активных атак (фальсификация данных и транзакций). Защита против подобных атак известна как аутентификация сообщений.

Сообщение, файл, документ или другое множество данных является подлинным (аутентичным), когда оно истинно и пришло из предполагаемого источника. Аутентификация сообщения есть процедура, которая позволяет взаимодействующим сторонам проверять, являются ли получаемые сообщения истинными. Двумя важными аспектами в этом вопросе являются проверка того, что содержимое сообщения не изменено и источник сообщения является подлинным. Также желательно проверять своевременность сообщения (доказательство того, что сообщение не воспроизведено и не имеет временных задержек) и последовательность сообщения в информационном потоке в целом.

1. Аутентификация с использованием стандартного шифрования

Возможно, выполнять аутентификацию, просто используя стандартное шифрование. Если только отправитель и получатель разделяют секретный ключ (как и должно быть), тогда только настоящий отправитель будет в состоянии зашифровать сообщение. Более того, если сообщение содержит код обнаружения ошибок и номер последовательности, то получатель будет уверен, что содержимое сообщения не менялось по пути следования, и оно получено в правильной последовательности. А если сообщение к тому же включает временной штамп, то получатель уверен, что сообщение не задерживалось по пути следования.

Аутентификация сообщений без шифрования

Во всех подобных методах генерируется так называемый аутентификационный признак (тэг), который затем добавляется к каждому сообщению при его передачи. Сообщение само по себе не шифруется и может быть прочтено получателем независимо от его функции аутентификации.

Из-за отсутствия шифрования не обеспечивается конфиденциальность сообщения. Так как стандартное шифрование обеспечивает аутентификацию и широко используется в многих готовых программных продуктах, почему бы просто не использовать его. При этом мы получим и конфиденциальность, и функцию аутентификации. Существует как минимум три ситуации, когда предпочтительна аутентификация без конфиденциальности:

Существует ряд приложений, в которых одно и то же сообщение передается широковещательным образом нескольким получателям. Дешевле и более надежно иметь только одного получателя, который будет ответственен за мониторинг подлинности сообщения. Таким образом, подобные сообщения должны передаваться в открытом виде с добавлением аутентификационного тэга. Ответственная система выполняет аутентификацию. Если производится нарушение, то другие системы оповещаются о нем путем получения сигнала предупреждения.

Другим возможным сценарием является обмен, в котором одна сторона сильно загружена и не имеет времени на дешифровку всех входящих сообщений. Аутентификация производится выборочно.

Аутентификация компьютерной программы является привлекательной возможностью. Компьютерная программа может выполняться без необходимости быть расшифрованной, что разгружает процессорное время. Тем не менее, если аутентификационный признак добавлен к программе, то он может проверяться, когда необходимо проверить ее целостность.

Код аутентификации сообщения

Одна из аутентификационных техник предполагает использование секретного ключа для генерации маленького блока данных, называемого аутентификационным кодом. Данный код добавляется к сообщению. Данная техника предполагает, что две взаимодействующие стороны, А и Б, разделяют секретный ключ КАБ. Когда А посылает сообщение Б, она вычисляет аутентификационный код как функцию сообщения и ключа. Затем сообщение и код передаются предполагаемому получателю. Получатель выполняет те же вычисления для генерации своего аутентификационного кода, а затем сравнивает его с полученным. Если только отправитель и получатель знают секретный ключ и если полученный код совпадает с вычисленным, то:

Получатель может быть уверен, что содержимое сообщения не изменено.

− Получатель может быть уверен, что сообщение получено от предполагаемого источника.

− Если сообщение включает номер последовательности, то получатель может быть уверен, что последовательность сообщений верна.

Для генерации аутентификационного кода может использоваться ряд алгоритмов. Национальное Бюро Стандартов США (публикация «Режимы операций алгоритма DES») рекомендует использовать DEA. DEA используется для генерации зашифрованной версии сообщения и только несколько последних бит этого шифра используются как код. Обычно это 16

или 32 битовый код.

Рисунок 1. показывает три способа аутентификации сообщения. Слепок сообщения может быть зашифрован с использованием стандартного шифрования (a). Сообщение также может быть зашифровано с использованием шифрования на основе общего ключа (b). Данный способ шифрования имеет два преимущества: наряду с аутентификацией он обеспечивает цифровую подпись; не требуется применения техники распределения ключей.

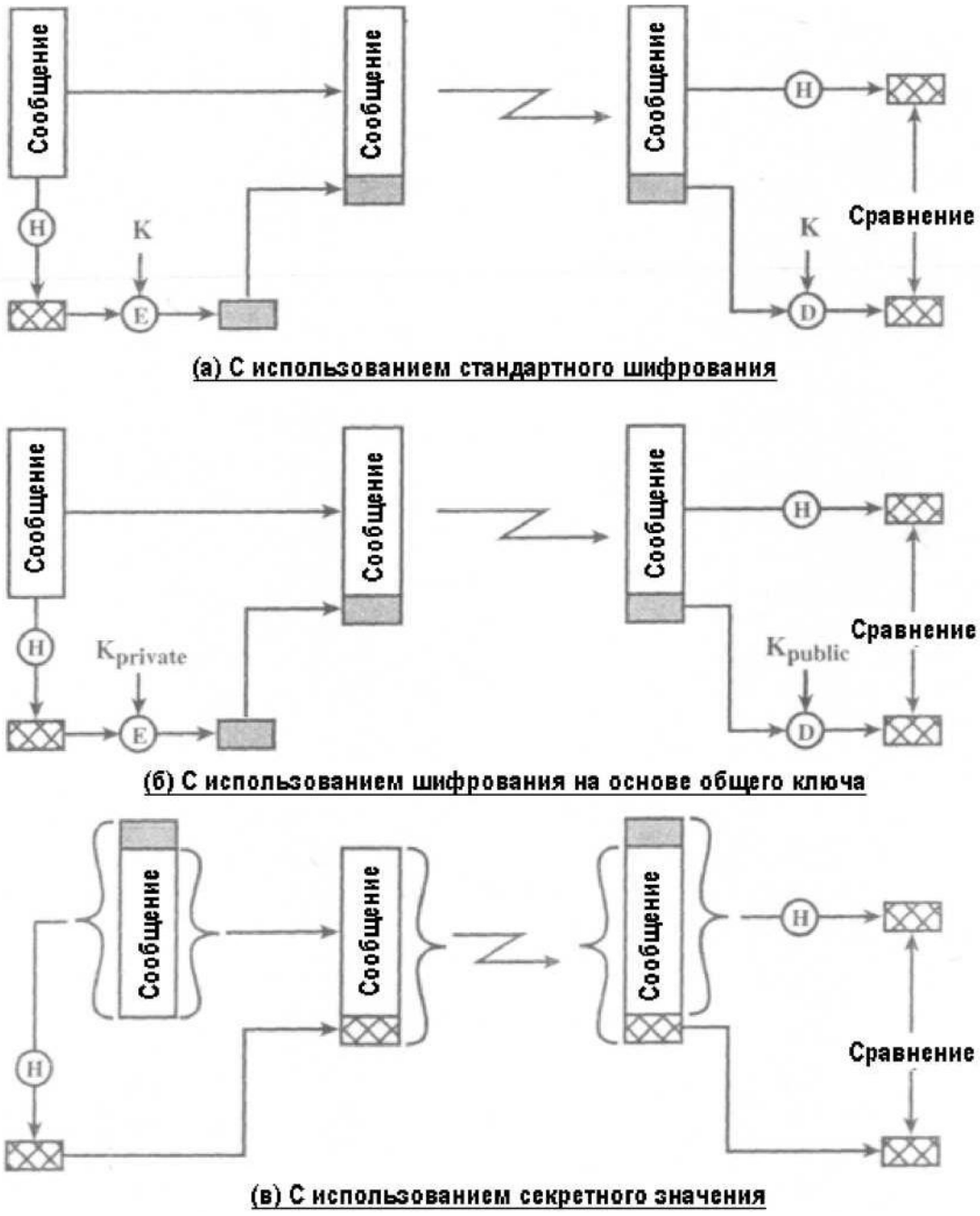


Рисунок 3.1 - Аутентификация сообщения с использованием односторонней кэш функции.

2. Односторонняя кэш функция

Одним из вариантов создания аутентификационного кода является односторонняя кэш функция. Кэш функция в качестве входных параметров берет сообщение M переменного размера и производит слепок H(M) сообщения, имеющий фиксированный размер. В отличие от предыдущей технологии (код аутентификации сообщения) кэш функция не использует секретный ключ. Для того, чтобы аутентифицировать сообщение, его слепок посылается вместе с самим сообщением получателю.

Данные два метода имеют преимущество над теми, которые шифруют все сообщение целиком, - требуется проделывать меньше вычислений. Тем не менее, остается интерес в разработке техники аутентификации, которая бы не применяла шифрование вообще. На это существует ряд причин:

− шифровальное ПО достаточно медленно;

шифровальное оборудование недешево. Существуют недорогие чипы для шифрования по DES, но стоимость растет, если необходимо внедрить подобные чипы во все узлы сети;

− шифровальное оборудование оптимизируется для обработки больших объемов данных. Для маленьких блоков, данных затрачивается больше времени на инициализацию и вызов функций;

− алгоритмы шифрования могут быть защищены патентами;

− алгоритмы шифрования могут быть предметом контроля над экспортом. Это относится к DES.

Рисунок 3.10в показывает технику, которая использует кэш функцию, но не использует шифрования для аутентификации сообщения. Предполагается, что две взаимодействующие стороны, А и Б, разделяют между собой общее секретное значение SAB. Когда А хочет послать сообщение Б, она вычисляет кэш функцию как конкатенацию секретной величины и сообщения: MDM = H(SAB || M). Затем А посылает Б сообщение [M || MDM]. Так как Б обладает величиной SAB, то он может проверить пришедшее сообщение. Так как секретная величина не посылается, то злоумышленник не может модифицировать сообщение в пути.

3. Безопасные кэш функции и HMAC

Одностороння кэш функция, или безопасная кэш функция, полезна не только для аутентификации сообщений, но и в цифровых подписях.

Требования к кэш функциям

Основное назначение кэш функции заключается в производстве «отпечатков пальцев» файла, сообщения или другого блока данных. Для того, чтобы быть применимой для аутентификации сообщений, кэш функция H должна обладать следующими свойствами:

H может быть применена к блоку данных любого размера.

H производит выходные данные фиксированного размера.

H(x) легко посчитать для любого заданного x.

Для любого заданного кода h невозможно найти такое x, что H(x) = h.

Для любого заданного блока x невозможно найти y Ф x так, что H(x) = H(y).

Невозможно найти любую пару (x, y) такую, что H(x) = H(y).

Первые три свойства являются требованиями для практического применения кэш- функции при аутентификации сообщений. Четвертое свойство является “односторонним”: легко сгенерировать код для заданного сообщения, но практически невозможно сгенерировать сообщение по заданному коду. Данное свойство является важным, если аутентификация включает использование секретного ключа. Сам секретный ключ не посылается. Тем не менее, если кэш-функция является не односторонней, то атакующий может легко раскрыть секретный ключ.

Пятое свойство гарантирует, что не возможно найти альтернативное сообщение с тем же самым кэш значением, что и для заданного сообщения. Это предохраняет от подделок.

− Стойкость ко взлому криптографических систем

Схема шифрования является безопасной с точки зрения вычислений, если шифр-текст, сгенерированный этой схемой отвечает одному или более условиям:

− стоимость взлома шифра превышает ценность зашифрованной информации;

− время взлома превышает полезное время жизни информации.

Проблема заключается в том, что очень трудно вычислить количество попыток, требуемых для удачного взлома шифра. Тем не менее, если предположить, что алгоритм шифрования не имеет никаких математических изъянов и, следовательно, для его взлома применима только брут-форс атака, то тогда мы можем сделать некоторые предположения относительно времени и стоимости взлома.

Суть брут-форс атаки заключается в том, что при взломе шифра перебираются все возможные ключи до тех пор, пока не будет получен нужный результат. В среднем необходимо перебрать половину ключей. Если множество ключей слишком большое, подобная задача становится практически невозможной. Таблица 3.5 показывает, как много времени требуется для перебора ключей различной длины.

Таблица 3.1 - Среднее время, требуемое для полного перебора всего множества ключей

Размер ключа (бит)	Количество ключей	Затрачиваемое время при скорости 1 ключ/мкс	Затрачиваемое время при скорости 106 ключей/мкс
32	232 = 4,3 * 109	231 мкс = 35,8 минут	2,15 миллисекунд
56	256 = 7,2 * 1016	255 мкс = 1142 лет	10 часов
128	2128 = 3,4 * 1038	2127 мкс = 5,4 * 1024 лет	5,4 * 1018 лет
168	2168 = 3,7 * 1050	2167 мкс = 5,9 * 1036 лет	5,4 * 1030 лет

Контрольные вопросы.

1. Когда используется аутентификация сообщений без шифрования;
2. Для чего нужен аутентификация сообщения;
3. Что такое односторонняя функция;
4. Расшифровать HMAC и рассказать принцип работы;
5. В чем заключается суть брут-форс атаки и как с ним борются.

Лекция 4. Механизмы обеспечения безопасности коммутируемых локальных сетей

Вопросы лекции:

1) Ограничение количества управляющих компьютеров.
2) Сегментация трафика (Traffic Segmentation).
3) Протокол IEEE 802.1 x.

Цель занятия ознакомить студентов различными механизмами обеспечения безопасности коммутируемых локальных сетей, а также рассказать как с помощью протокола IEEE 802.1 x. допустить для управления коммутатора только аутентифицированных пользователей.

1. Ограничение количества управляющих компьютеров

Современные коммутаторы позволяют задать конкретные компьютеры (IP-адреса), с которых будет происходить настройка данных коммутаторов по Web или Telnet-инерфейс сам или через протокол SNMP. Попытка прочих компьютеров подключиться к коммутаторам для управления ими будет отклонена.

Настройка безопасности индивидуального порта

Данная функция позволяет:

− заблокировать дальнейшее обновление таблицы коммутатора - если конфигурирование Вашей сети больше не изменятся, Вы блокируете таблицу коммутации, и коммутатор будет просто отбрасывать все пакеты, которые поступают с неизвестных адресов. Причём данное действие можно выбрать для конкретных портов. При этом записи в таблице коммутации не будут удаляться по тайм-ауту;

− задать максимальное количество MAC-адресов для привязки к конкретному порту.

Фильтрация MAC-адресов

Дополнительно можно настроить коммутатор так, чтобы он не принимал пакеты с определенным MAC-адресом (как получателя, так и отправителя). Для этого служит таблица фильтрации трафика (Filtering Table). Фактически данная таблица является «чёрным списком». После получения пакета коммутатор считывает оба аппаратных адреса, как получателя, так и отправителя. Если хотя бы один из них содержится в таблице фильтрации, то пакет отбрасывается.

Технология фильтрации IP-MAO Binding

Основное назначение данной технологии - это ограничить доступ к коммутатору определённого количества компьютеров. Для этого индивидуально для каждого желаемого порта создаётся таблица соответствия IP- и MAC-адресов. Далее коммутатор будет пропускать только пакеты с указанными MAC-адресами и только в том случае, если истинно соответствие IP- и MAC-адреса.

Существенные отличия данной технологии от технологии фильтрации MAC-адресов:

− фильтрация работает на всех портах, а в данной технологии можно настраивать каждый порт в отдельности;

− фильтрация содержит «чёрный» список, то есть работает по принципу: не пропускать те пакеты, MAC-адреса которых содержатся в таблице фильтрации. Данная технология, наоборот, содержит «белый» список, то есть пропускает только те пакеты, MAC-адреса которых содержатся в созданных списках;

− в отличие от фильтрации технология IP-MAC Binding проверяет не только MAC- адрес источника пакета, но и его IP-адрес.

Списки контроля доступа (Access Control Lists)

Списки контроля доступа обеспечивают ограничение прохождения трафика через коммутатор. Фактически технология ACL реализует на коммутаторах 3-го уровня полноценный фильтр пакетов. Приём пакетов или отказ в приёме основывается на определённых признаках, которые содержит пакет:

− IP- и MAC-адреса источника и приёмника;
− номер VLAN;
− номер порта TCP или UDP;
− тип ICMP-сообщения.

Ключевым понятием в данной технологии является понятие профиля доступа - это набор признаков, который содержит пакет, с определёнными значениями. Каждый профиль доступ имеет свой уникальный номер в пределах коммутатора. Пример профилей доступа:

1: <VLAN = Yes, Source MAC = 00-01-08-DE-FA-10, Destination MAC = 00-01-07-DE-FA-10>

2: <VLAN = Yes, Source IP = 192.168.3.1, Destination IP = 192.168.10.2, TCP port = Yes>

Помимо приведенных примеров существуют также контекстно-зависимые профили. Основное отличие контекстно-зависимых профилей от всех остальных заключается в том, что в них признаки задаются смещением относительно начала кадра Ethernet. Например, чтобы указать MAC-адрес источника пакета в контекстно-зависимом профиле необходимо указать смещение Offset = 6 байтам и далее значение MAC-адреса, так как данный адрес располагается в заголовке кадра Ethernet с 7 по 12 байт включительно.

Профиль является всего лишь образцом (некоторым эталоном), на

основе которого создаются правила. Правила представляют собой профиль, заполненный конкретными значениями признаков и содержащий действие, которое необходимо выполнить над пакетом, если он попадает под эти признаки. Обычно действие - это удалить или продвинуть пакет. Для каждого профиля может быть создано более одного правила. Например, для вышеприведённого профиля с ID=2 могут быть созданы следующие правила:

Accept: <VLAN = Default, Source IP = 192.168.3.1, Destination IP = 192.168.10.2, TCP port = 514>

Deny: <VLAN = Marketing, Source IP = 192.168.3.1, Destination IP = 192.168.10.2, TCP port = 8080>

При поступлении пакета на коммутатор профили доступа применяются последовательно, в порядке возрастания их номеров. Пакет проверяется на соответствие условиям, указанным во всех правилах профиля, начиная с первого профиля. Если правило подходит, пакет в соответствие с действием обработки либо принимается, либо отбрасывается и дальше не проверяется. Если пакет не попадает ни под одно правило профиля, то пакет проверяется по правилам следующего профиля. Если не один профиль не подходит, то применяется политика по умолчанию - разрешающая или запрещающая прохождение трафика.

Очень важно правильно расставлять профили и правила внутри профилей потому, что если пакет попадает хотя бы под одно запрещающее правило, то он удаляется из коммутатора и дальше не проверяется на соответствие другим правилам, даже если среди них есть правила, которые могли бы его пропустить дальше.

2. Сегментация трафика (Traffic Segmentation)

Сегментация трафика служит для разграничения портов на Канальном уровне. Данная функция позволяет настраивать порты или группу портов таким образом, чтобы они были изолированы друг от друга, но в то же время имели доступ к разделяемым портам, используемым для подключения серверов или магистрали сети провайдера. Очень важной является возможность одновременного использования данной функции с виртуальными сетями (VLAN), что позволяет производить дальнейшее разграничение прав.

Данная технология схожа с технологией VLAN, но является более ограниченной по функциональности. К преимуществам технологии можно отнести:

простота настройки;

более низкая загрузка процессора коммутатора по сравнению с VLAN.

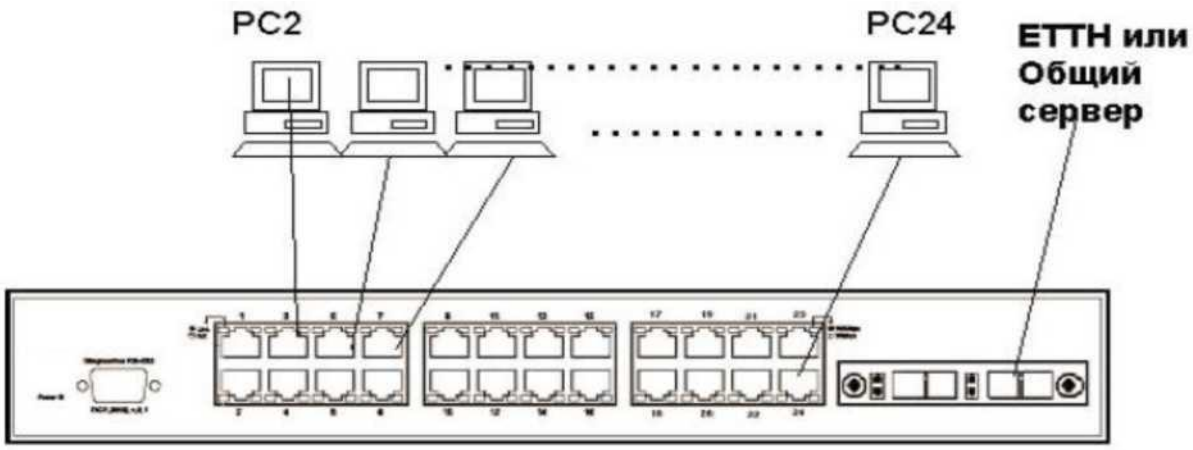


Рисунок 4.1 - коммутатор

Рассмотрим пример (рисунок 4.1). Все компьютеры имеют доступ к общему порту, но не имеют доступа к друг другу на Канальном уровне. Это решение может быть использовано

− в проектах ETTH (Ethernet To The Home, Ethernet до дома) для изоляции компьютеров конечных пользователей;

− для предоставления доступа к общему серверу.

3. Протокол IEEE 802.1 x

Протокол IEEE 802.1x является механизмом безопасности, обеспечивающим аутентификацию и авторизацию пользователей и тем самым ограничивающим доступ проводных или беспроводных устройств к локальной сети. Работа протокола базируется на клиент- серверной модели контроля доступа (рисунок 4.2). В качестве сервера аутентификации используется RADIUS-сервер. При этом весь процесс аутентификации пользователя производится в проводных сетях на основе протокола EAPOL (Extensible Authentication Protocol over LAN), в беспроводных - на основе протокола EAPOW (Extensible Authentication Protocol over Wireless).

Рисунок 4.2 – Взаимная аутентификация

До тех пор, пока клиент не будет аутентифицирован, протокол IEEE 802.1x будет пропускать через сетевой порт только трафик протокола EAPOL. После успешной аутентификации обычный трафик будет пропускаться через

порт. Работа протокола IEEE 802.1x основывается на трёх компонентах (рисунок 4.2), каждая из которых подробно рассмотрена в следующем разделе.

Роли устройств

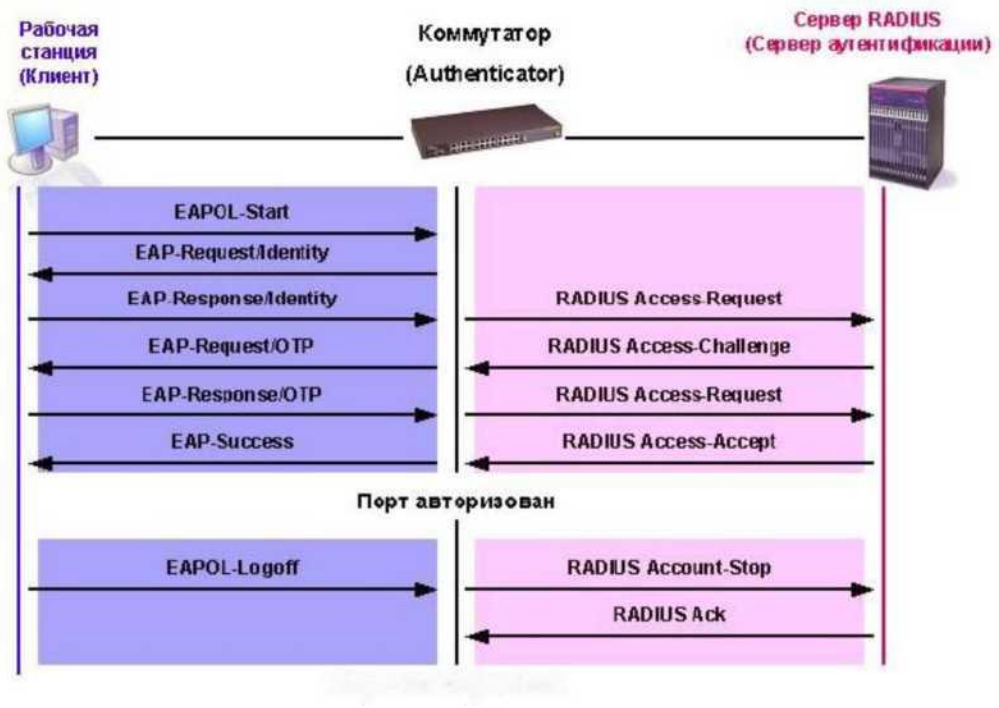
Клиент - это рабочая станция, которая запрашивает доступ к локальной сети и сервисам коммутатора и отвечает на запросы коммутатора. На рабочей станции должно быть установлено клиентское ПО, реализующее протокол 802.1x (в ОС Microsoft Windows XP данное ПО является встроенным).

Сервер аутентификации выполняет фактическую аутентификацию клиента, проверяя подлинность клиента и информируя коммутатор, предоставлять или нет клиенту доступ к локальной сети.

Коммутатор (также называется аутентификатор) управляет физическим доступом к сети, основываясь на статусе аутентификации клиента. Коммутатор работает как посредник между клиентом и сервером аутентификации, получая запрос на проверку подлинности от клиента, проверяя данную информацию при помощи сервера аутентификации, и пересылая ответ клиенту. ПО коммутатора включает клиента RADIUS, который отвечает за инкапсуляцию и деинкапсуляцию кадров EAP и взаимодействие с сервером аутентификации.

Процесс аутентификации

Инициировать процесс аутентификации может коммутатор или клиент. Клиент инициирует аутентификацию, посылая кадр EAPOL-start, который вынуждает коммутатор отправить ему запрос на идентификацию. Когда клиент отправляет EAP - ответ со своей идентификацией, коммутатор начинает играть роль посредника, передающего кадры EAP между клиентом и сервером аутентификации до успешной или неуспешной аутентификации. Если аутентификация завершилась успешно, порт коммутатора становится авторизованным.



Порт не авторизован

Рисунок 4.3 - Временная диаграмма аутентификации клиента в сети.

Временная диаграмма обмена EAP-кадрами зависит от используемого метода аутентификации. На рисунке 4.3 показана схема обмена, инициируемая клиентом, использующая метод аутентификации с использованием одноразовых паролей (One Time Password, OTP) сервером RADIUS.

Состояние портов коммутатора

Состояние порта коммутатора определяется тем, получил или не получил клиент право доступа к сети. Первоначально порт находится в неавторизованном состоянии. В этом состоянии он запрещает прохождение всего входящего и исходящего трафика за исключением пакетов протокола IEEE 802.1x. Когда клиент аутентифицирован, порт переходит в авторизованное состояние, позволяя передачу любого трафика от него.

Возможны варианты, когда клиент или коммутатор не поддерживают протокол IEEE 802.1x. Если клиент, который не поддерживает протокол IEEE 802.1x, подключается к неавторизованному порту, коммутатор посылает клиенту запрос на аутентификацию. Поскольку в этом случае клиент не ответит на запрос, порт останется в неавторизованном состоянии и клиент не получит доступ к сети.

В другом случае, когда клиент с поддержкой протокола IEEE 802.1x подключается к порту, на котором не запущен протокол IEEE 802.1x, клиент начинает процесс аутентификации, посылая кадр EAPOL-start. Не получив ответа, клиент посылает запрос определённое количество раз. Если после этого ответ не получен, клиент, считая, что порт находится в авторизованном

состоянии начинает посылать кадры.

В случае, когда и клиент и коммутатор поддерживают протокол IEEE 802.1x, при успешной аутентификации клиента, порт переходит в авторизованное состояние и начинает передавать все кадры клиента. Если в процессе аутентификации возникли ошибки, порт остаётся в неавторизованном состоянии, но аутентификация может быть восстановлена. Если сервер аутентификации не может быть достигнут, коммутатор может повторно передать запрос. Если от сервера не получен ответ после определённого количества попыток, то в доступе к сети будет отказано из-за ошибок аутентификации.

Когда клиент завершает сеанс работы, он посылает сообщение EAPOL-logoff, переводящее порт коммутатора в неавторизованное состоянии. Если состояние канала связи порта переходит из активного (up) в неактивное (down), то порт также возвращается в неавторизованное состояние.

Методы контроля доступа при использовании протокола IEEE 802.1x

Протокол IEEE 802.1x предоставляет два метода контроля доступа к сети:

− На основе портов (Port-Based Access Control). При использовании данного метода достаточно, чтобы только один любой пользователь, подключенный к порту коммутатора, был авторизован. Тогда порт перейдёт в авторизованное состояние и доступ к сети получат любые пользователи, подключенному к данному порту.

− На основе MAC-адресов (MAC-Based Access Control). При использовании данного метода при аутентификации также учитывается MAC-адрес клиента, подключенного к порту, и порт авторизуется только для клиента с конкретным MAC-адресом.

Контроль доступа на основе портов

Изначально протокол IEEE 802.1x разрабатывался с учётом того, что к порту коммутатора подключено не более одного устройства (рисунок 4.4). Как только устройство успешно проходило процедуру аутентификации, порт переходил в авторизованное состояние и далее пропускал весь трафик до тех пор, пока не наступало событие, которое обратно переводило его в неавторизованное состояние. Следовательно, если порт коммутатора подключен не к одному устройству, а к сегменту локальной сети, то успешная аутентификация любого устройства из этого сегмента открывает доступ в сеть всем остальным устройствам из сегмента. Естественно, это является серьёзной проблемой с точки зрения безопасности.

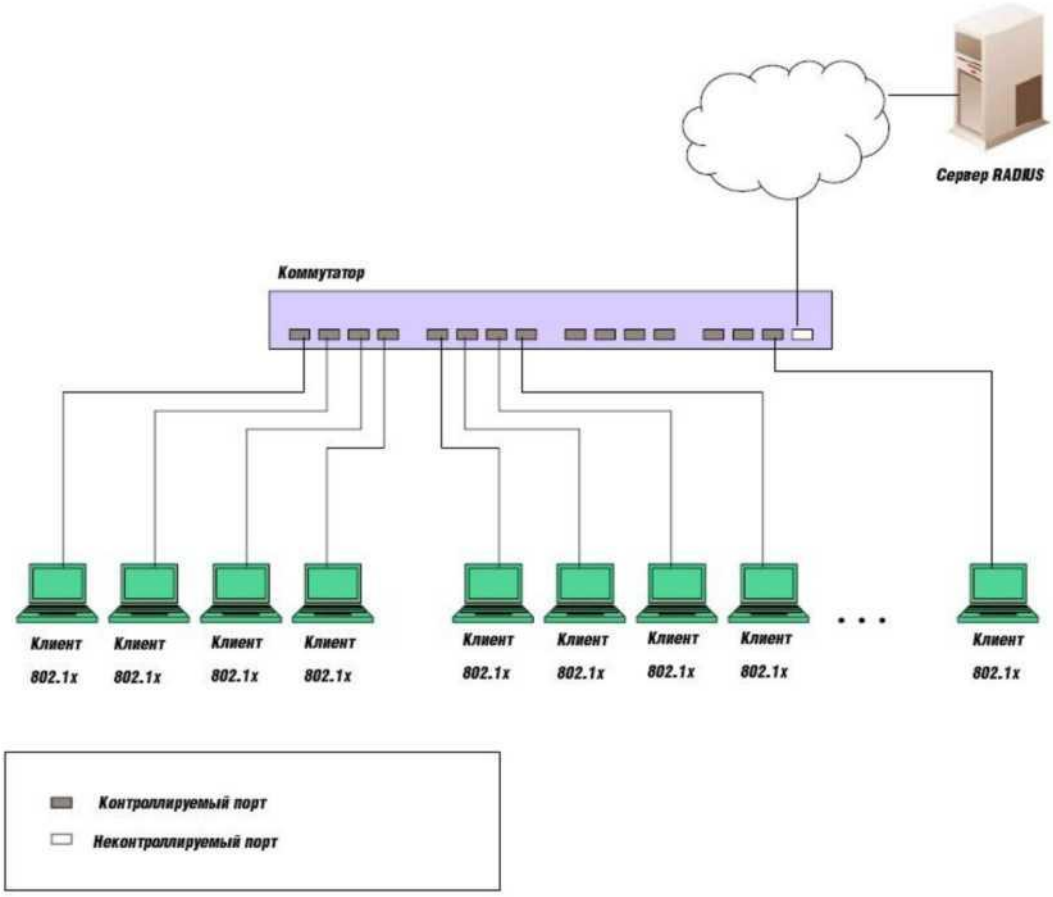


Рисунок 4.4 – Контроль доступа на основе портов

Контроль доступа на основе MAC-адресов

Для того, чтобы успешно использовать протокол IEEE 802.1x в распределённых локальных сетях, необходимо создавать логические порты - по одному логическому порту на каждое устройство, подключенное к физическому порту. Таким образом, физический порт представляет собой множество логических портов, каждый из которых независимо контролирует отдельное устройство-клиента с точки зрения аутентификации и авторизации. Принадлежность устройства к определённому логическому порту осуществляется на основе MAC-адреса устройства (рисунок 4.5). Таким образом, устраняется проблема безопасности доступа множества устройств через один физический порт коммутатора.

Контрольные вопросы.

1. Какими способами можно ограничить количество компьютеров для управления коммутаторов;
2. Для чего нужен Radius сервер;
3. Что такое One Time Password, OTP;
4. По каким параметрам можно создать списки контроля доступа;
5. Как работает IEEE 802.1x.

Лекция 5. Виртуальные сети (Virtual LAN)

Вопросы лекции:

1) Виртуальная ЛВС.

2) Сети на базе портов (Port-based VLANs).

3) Построение виртуальных сетей 802.1 Q на одном коммутаторе.

Цель занятия ознакомить студентов для чего нужен виртуальная локальная сеть и в каких устройствах можно создавать и каким образом можно обеспечить безопасность коммутируемых локальных сетей.

1. Виртуальная ЛВС

Виртуальная ЛВС (VLAN, Virtual LAN) - логическая группа компьютеров в пределах одной реальной ЛВС, за пределы которой не выходит любой тип трафика (широковещательный [broadcast], многоадресный [multicast] и одноадресный [unicast]). За счет использования VLAN администратор сети может организовать пользователей в логические группы независимо от физического расположения рабочих станций этих пользователей.

Основные цели введения виртуальных сетей в коммутируемую среду:

− повышение полезной пропускной способности сети за счет локализации широковещательного (broadcast) трафика в пределах виртуальной сети;

− повышения уровня безопасности сети за счет локализации одноадресного (unicast) трафика в пределах виртуальной сети;

− формирование виртуальных рабочих групп из некомпактно (в плане подключения) расположенных узлов;

− улучшение соотношения цены/производительности по сравнению с применением маршрутизаторов.

Классический пример, отражающий суть виртуальных сетей, приведен на рисунке 4.7. К одному коммутатору гипотетической фирмы подключены как машины бухгалтеров, так и машины инженеров. При этом совершенно нет никакой необходимости во взаимодействие машин данных двух групп сотрудников. Поэтому машины бухгалтеров выделяются в одну виртуальную сеть, а машины инженеров в другую. При этом весь трафик (широковещательный, многоадресный и одноадресный) будет ограничен пределами своей виртуальной сети. Более того, повысится безопасность сети. Например, если на машине бухгалтера появится вирус «червь», то максимум он сможет заразить только машины бухгалтеров.

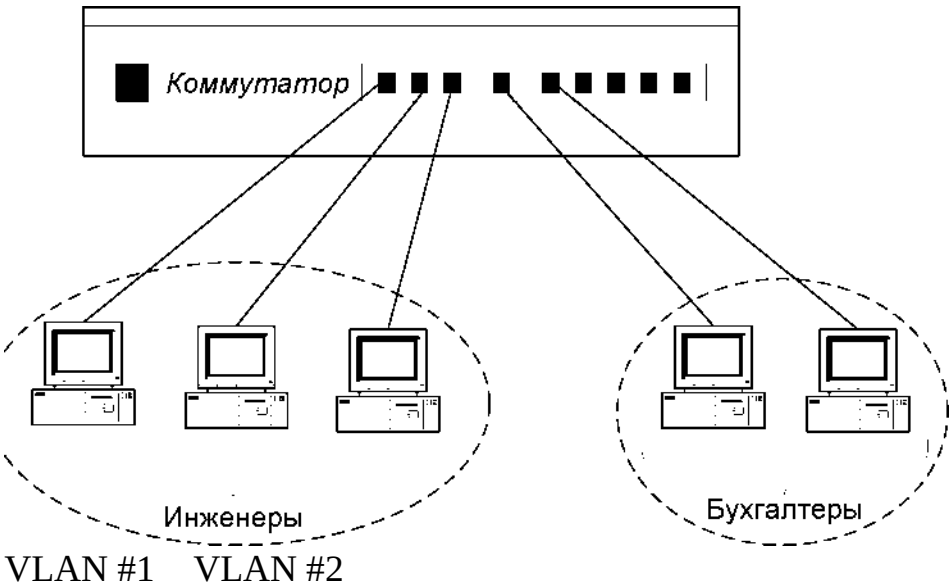


Рисунок 5.1 - Пример использования технологии виртуальных сетей.

Сегодня существует достаточно много вариантов реализации VLAN. Простые варианты VLAN представляют собой набор портов коммутатора, более сложные реализации позволяют создавать группы на основе других критериев. В общем случае возможности организации VLAN тесно связаны с возможностями коммутаторов.

Сети на базе MAC-адресов (MAC-based VLANs)

Данный тип виртуальных сетей группирует устройства на основе их MAC-адресов. Для получения доступа в виртуальную сеть, устройство должно иметь MAC-адрес, который содержится в списке адресов данной виртуальной сети. Помимо прочего, отличительной особенностью данного типа виртуальных сетей является то, что они ограничивают только широковещательный трафик. Отсюда вытекает их название - широковещательные домены на базе MAC-адресов. Теоретически один MAC-адрес может являться членом нескольких широковещательных доменов, на практике данная возможность определяется функциональностью конкретной модели коммутатора.

Настройка виртуальной сети на основе MAC-адресов может отнять много времени. Представьте себе, что вам потребуется связать с VLAN адреса 1000 устройств. Кроме того, MAC-адреса "зашиты" в оборудование и может потребоваться много времени на выяснение адресов устройств в большой, территориально распределенной сети. Более того, существуют проблемы безопасности при работе с сетью на базе MAC-адресов. Злоумышленник может узнать MAC-адрес компьютера, входящего в ту или иную VLAN, назначить его своей сетевой карте (как минимум, это можно сделать стандартными средствами MS Windows XP) и успешно подключиться к сети.

С другой стороны, широковещательные домены на базе MAC-адресов позволяют физически перемещать станцию, позволяя, тем не менее,

оставаться ей в одном и том же широковещательном домене без каких-либо изменений в настройках конфигурации. Это удобно в сетях, где станции часто перемещают, например, где люди, использующие ноутбуки, постоянно подключаются в разных частях сети.

2. Сети на базе портов (Port-based VLANs)

Устройства связываются в виртуальные сети на основе портов коммутатора, к которым они физически подключены. То есть каждой порт коммутатора включается в одну или более виртуальных сетей. К достоинствам данного типа виртуальных сетей можно отнести высокий уровень безопасности и простоту в настройке. К недостаткам можно отнести статичность данного типа виртуальных сетей. То есть при подключении компьютера к другому порту коммутатора необходимо каждый раз изменять настройки VLAN.

Сеть на базе маркированных кадров (IEEE 802.1Q VLANs)

В отличие от двух предыдущих типов виртуальных сетей VLAN на основе маркированных кадров могут быть реализованы на двух и более коммутаторах. В заголовок каждого кадра Ethernet вставляется маркер, который идентифицирует членство компьютера в определенной VLAN. На рисунке 4.8 показан пример такой VLAN.

Механизмы добавления идентифицирующего маркера в заголовок кадра Ethernet

Маркеры с номером VLAN в виртуальных сетях 802.1Q могут быть добавлены:

− явно, если сетевые карты поддерживают стандарт IEEE 802.1Q, и на этих картах включены соответствующие опции, то исходящие кадры Ethernet от этих карт будут содержать маркеры идентификации;

− неявно, если сетевые адаптеры, подключенные к этой сети, не поддерживают стандарт IEEE 802.1Q, то добавление маркеров выполняется на коммутаторе на основе группировки по портам.

Предположим, что некоторые порты коммутатора сгруппированы в VLAN. Коммутатор с поддержкой IEEE 802.1Q будет добавлять маркер к входящим на этот порт кадрам Ethernet с соответствующим ID VLAN. Но эти маркеры будут удалены коммутатором из исходящих с этих же портов кадров, чтобы конечные компьютеры могли разобрать заголовок кадра Ethernet.

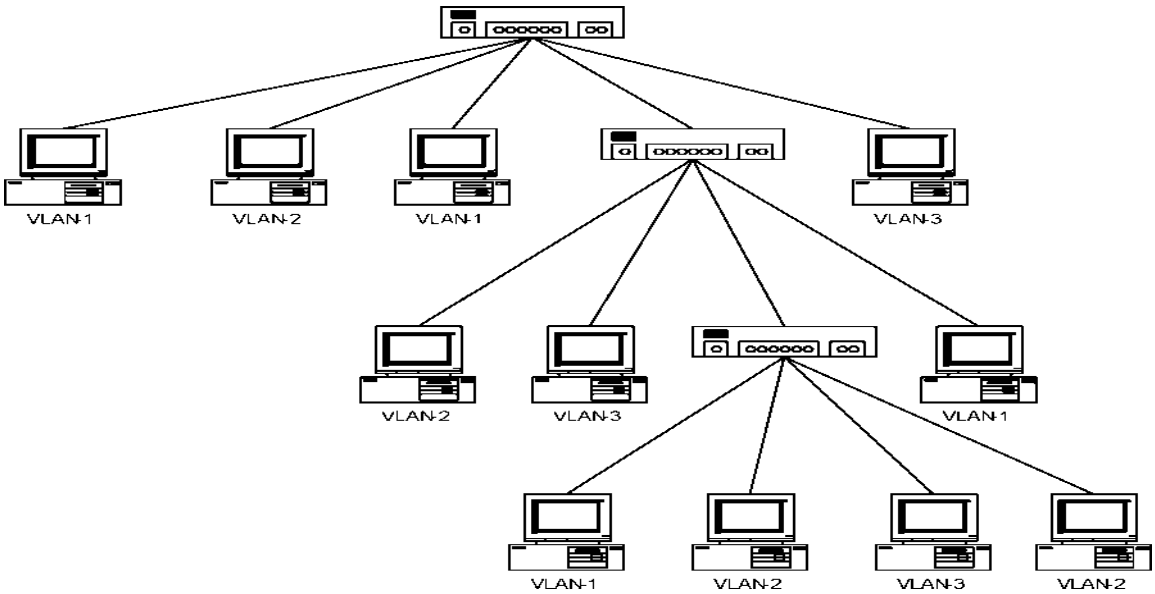


Рисунок 5.2 - Виртуальная сеть на основе нескольких коммутаторов с использованием стандарта IEEE 802.1Q.

Если идентификация VLAN маркерами протокола 802.1Q была осуществлена обоими способами - явно и неявно, входящие кадры к портам коммутатора могут состоять из обоих типов кадров (с маркерами и без). В этой ситуации к неотмеченным входящим кадрам будут добавляться маркеры. В то время как маркированные кадры уже поддерживают членство в явно определенной виртуальной сети. Способность VLAN 802.1Q извлечения маркеров из заголовков кадров Ethernet позволяет VLAN работать с существующими коммутаторами и сетевыми адаптерами, которые не распознают маркеры. Способность добавления маркеров позволяет VLAN распространяться через множество 802.1 Q- совместимых коммутаторов.

Компоненты конфигурации виртуальной сети на основе тэгов

Необходимо понять суть двух ключевых переменных для настройки VLAN 802.1Q:

- Port VLAN ID (PVID, идентификатор номера порта);
- VLAN ID (VID, идентификатор номера VLAN).

Обе переменные назначаются для одного и того же порта коммутатора, но между ними имеется существенная разница.

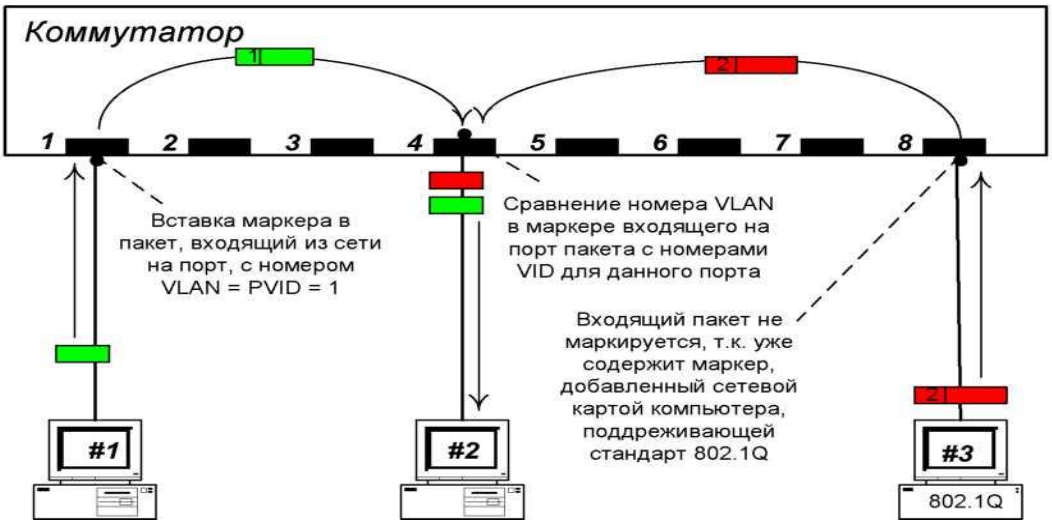
Пользователь может назначить ТОЛЬКО ОДИН номер PVID для каждого порта. Номер PVID определяет, к какой VLAN принадлежит данный порт. Когда ИЗ СЕТИ на порт коммутатора поступает НЕМАРКЕРОВАННЫЙ пакет, то принадлежность данного пакета к определенной VLAN осуществляется как раз на основе номера PVID порта. Номер PVID вставляется в заголовок кадра Ethernet и идентифицирует членство пакета в одной определенной VLAN. Необходимо отметить важный момент: если пакет уже содержит маркер (заданный явно сетевой картой компьютера или заданный неявно на другом коммутаторе), то повторная маркировка пакета,

поступающего из вне на какой-либо порт коммутатора, НЕ производится.

С другой стороны, можно задать МНОЖЕСТВО идентификаторов VID для одного и того же порта. Данные идентификаторы определяют, пакеты каких виртуальных сетей может принимать данный порт. Данная проверка осуществляется, когда пакет УЖЕ ПОЛУЧЕН ИЗ СЕТИ и МАРКИРОВАН, во время процедуры перенаправления пакета с порта на порт ВНУТРИ коммутатора.

3. Построение виртуальных сетей 802.1 Q на одном коммутаторе.

Рассмотрим следующую модель (рисунок 3), которая помогает понять принцип работы виртуальных сетей 802.1Q. В данном примере порт #1 является членом VLAN #1, а порты #4 и #8 являются членами VLAN #2. При этом порт #1 может принимать входящие пакеты только из своей VLAN, то есть только из первой VLAN. Порт #4 может принимать входящие пакеты из обоих VLAN (первой и второй). Порт #8, также как первый порт, может принимать входящие пакеты только из своей VLAN (второй). В результате полноценно функционировать в данной сети смогут только машины #1 #2 и #2 #3. Машины #1 и #3 не смогут обмениваться информацией, так как порт #4 может принимать пакеты с порта #1, но не наоборот.



PVID	Номер	VID	Номера
1	1	1	1,4
2	4	2	4,8
2	8	–	

Рисунок 5.3 - Модель функционирования виртуальной сети 802.1 Q на одном коммутаторе.

Построение виртуальных сетей 802.1 Q на нескольких коммутаторах

Как уже отмечалось, виртуальные сети 802.1 Q могут быть построены на

нескольких коммутаторах и охватывать всю локальную сеть. При этом все коммутаторы сети должны поддерживать стандарт 802.1Q. Необходимо понять следующие важные термины:

Маркировка (Tagging) - процесс вставки информации о виртуальной сети 802.1Q в заголовок кадра Ethernet. Порт, для которого включен режим маркировки, будет оставлять без изменения заголовок ИСХОДЯЩЕГО С КОММУТАТОРА В СЕТЬ пакета Ethernet, то есть будет оставлять в нем маркер 802.1 Q. Подобная возможность необходима для связи двух устройств, поддерживающих стандарт 802.1Q.

Демаркировка (Untagging) - процесс удаления информации о виртуальной сети 802.1Q из заголовка кадра Ethernet. Порт, для которого включен режим демаркировки, будет удалять маркер из заголовка пакета, ИСХОДЯЩЕГО С КОММУТАТОРА В СЕТЬ. Подобная возможность необходима для связи коммутатора и устройства, не поддерживающего стандарт 802.1Q.

Входящий порт (Ingress Port) - порт коммутатора, на который поступают пакеты из сети. Для каждого порта можно настроить фильтр Ingress Filter. Если этот фильтр отключен (состояние Disabled) то порт функционирует в обычном режиме, то есть проверка пакета на принадлежность какой-либо VLAN (сравнение номера VLAN пакета с номерами VID порта) производится при поступлении пакета на данный порт с другого порта ВНУТРИ коммутатора (как это показано на рисунке 4.9). При включении фильтра (состояние Enabled) данная проверка ДОПОЛНИТЕЛЬНО производится для любого маркированного пакета, входящего на данный порт ИЗ СЕТИ.

Исходящий порт (Egress Port) - порт коммутатора, с которого пакеты выходят в сеть и при этом необходимо принять решение о маркировке/демаркировке пакета.

С учетом всего ниже приведена расширенная модель функционирования виртуальной сети 802.1 Q на основе нескольких коммутаторов (рисунок 4.10).

Преимущества виртуальных сетей 802.1Q

Дополнительное преимущество VLAN 802.1Q заключается в том, что можно изменять топологию сети без физического перемещения станций или изменения кабельных соединений. Станции можно назначить другую VLAN и, соответственно, общаться и совместно использовать ресурсы с членами в новой VLAN просто изменив настройки порта с одной VLAN (например, VLAN отдела продаж) на другую (VLAN отдела маркетинга). Таким образом, VLAN обеспечивают гибкость при перемещениях, изменениях и наращивании сети. Следует отметить, что в современных коммутаторах поддерживается единственный тип VLAN - тэгированные виртуальные сети.

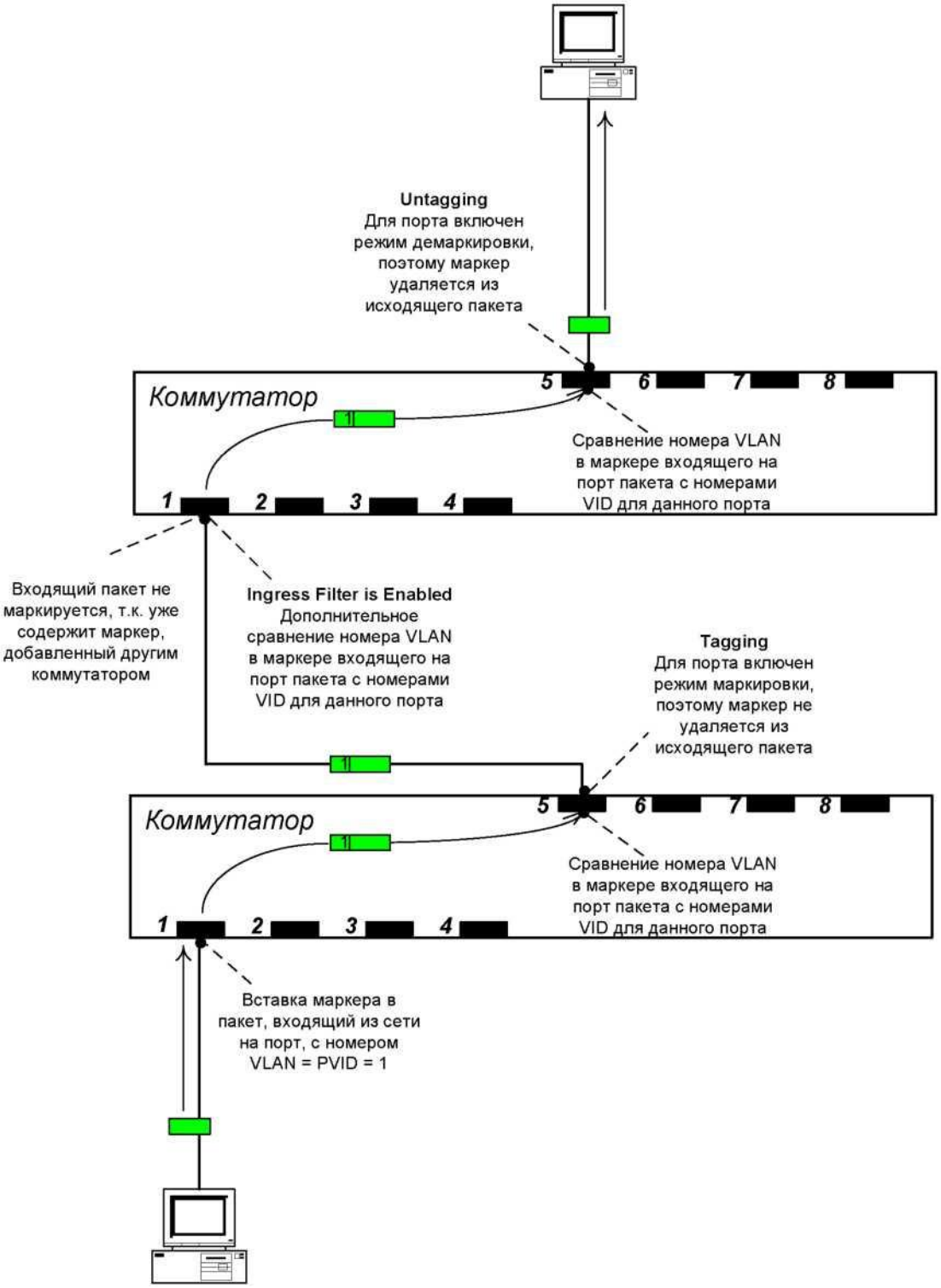


Рисунок 5.4 - Модель функционирования vlan 802.1q
на нескольких коммутаторах.

Контрольные вопросы.

1. Сколько PVID может назначить пользователь для каждого порта;
2. Какое максимальное число виланов можно создать;
3. Что чего нужен не тегированные порты;
4. В каких случаях используется тегированные порты;

5. Для конфигурации виртуальной сети на основе тэгов какие компоненты нужны.

Лекция 6. Механизмы обеспечения безопасности беспроводных локальных сетей

Вопросы лекции:

1) Классификация механизмов безопасности в сетях Wi-Fi.

2) Блочное шифрование.

3) Стандарт сети 802.11 i с повышенной безопасностью (WPA2).

Цель занятия ознакомить студентов различными механизмами обеспечения безопасности в сетях Wi-Fi такими как WEP (IEEE 802.11), WPA, WPA2 их преимущества и недостатков.

1. Классификация механизмов безопасности в сетях Wi-Fi

В современных беспроводных Wi-Fi сетях имеется следующий набор механизмов, обеспечивающих безопасность работы пользователя и конфиденциальность передаваемой информации. Данный набор представлен на рисунке 5.1.

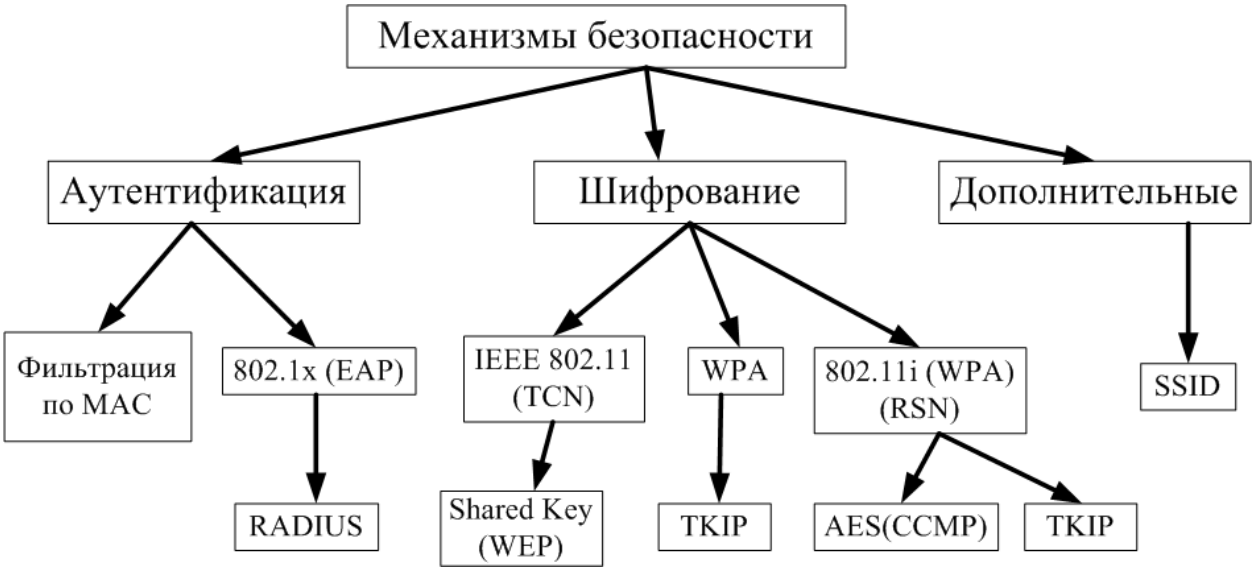


Рисунок 6.1 - Классификация механизмов безопасности в Wi-Fi сетях.

Механизмы шифрования

Механизм шифрования WEP (IEEE 802.11)

Шифрование WEP (Wired Equivalent Privacy - секретность на уровне проводной связи) основано на алгоритме RC4 (Rivest's Cipher v.4 - код Ривеста), который представляет собой симметричное потоковое шифрование. Как было отмечено ранее, для нормального обмена пользовательскими данными ключи шифрования у абонента и точки радиодоступа должны быть идентичными.

Ядро алгоритма состоит из функции генерации ключевого потока. Эта функция генерирует последовательность битов, которая затем объединяется с открытым текстом посредством суммирования по модулю два. Дешифрация состоит из регенерации этого ключевого потока и суммирования его с шифрограммой по модулю два для восстановления исходного текста. Другая главная часть алгоритма - функция инициализации, которая использует ключ переменной длины для создания начального состояния генератора ключевого потока.

RC4 - фактически класс алгоритмов, определяемых размером его блока. Этот параметр n является размером слова для алгоритма. Обычно, n = 8, но в целях анализа можно уменьшить его. Однако для повышения уровня безопасности необходимо задать большее значение этой величины. Внутреннее состояние RC4 состоит из массива размером 2n слов и двух счетчиков, каждый размером в одно слово. Массив известен как S-бокс, и далее он будет обозначаться как S. Он всегда содержит перестановку 2n возможных значений слова. Два счетчика обозначены через i и j. Алгоритм инициализации RC4 приведен ниже. Этот алгоритм использует ключ, сохраненный в Key и имеющий длину l байт. Инициализация начинается с заполнения массива S, далее этот массив перемешивается путем перестановок, определяемых ключом

Эффективность: WEP легко реализовать.

Открытость.

Использование WEP-шифрования не является обязательным в сетях стандарта IEEE 802.11.

Для непрерывного шифрования потока данных используется потоковое и блочное шифрование.

Потоковое шифрование

При потоковом шифровании выполняется побитовое сложение по модулю 2 (функция "исключающее ИЛИ", XOR) ключевой последовательности, генерируемой алгоритмом шифрования на основе заранее заданного ключа, и исходного сообщения. Ключевая последовательность имеет длину, соответствующую длине исходного сообщения, подлежащего шифрованию (рисунок 5.2).

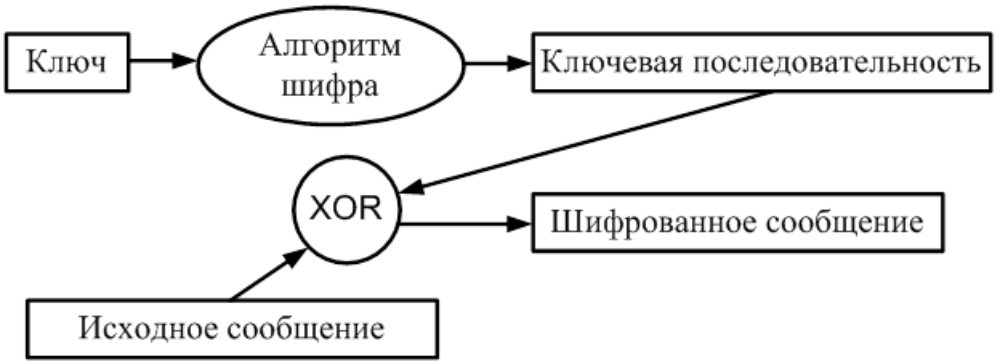


Рисунок 6.2 - Потоковое шифрование.

2. Блочное шифрование

Блочное шифрование работает с блоками заранее определенной длины, не меняющейся в процессе шифрования. Исходное сообщение фрагментируется на блоки, и функция XOR вычисляется над ключевой последовательностью и каждым блоком. Размер блока фиксирован, а последний фрагмент исходного сообщения дополняется пустыми символами до длины нормального блока (рисунок 5.3). Например, при блочном шифровании с 16-байтовыми блоками исходное сообщение длиной в 38 байтов фрагментируется на два блока длиной по 16 байтов и 1 блок длиной 6 байтов, который затем дополняется 10 байтами пустых символов до длины нормального блока.

Потоковое шифрование и блочное шифрование используют метод электронной кодовой книги (ECB). Метод ECB характеризуется тем, что одно и то же исходное сообщение на входе всегда порождает одно и то же зашифрованное сообщение на выходе. Это потенциальная брешь в системе безопасности, ибо сторонний наблюдатель, обнаружив повторяющиеся последовательности в зашифрованном сообщении, в состоянии сделать обоснованные предположения относительно идентичности содержания исходного сообщения.

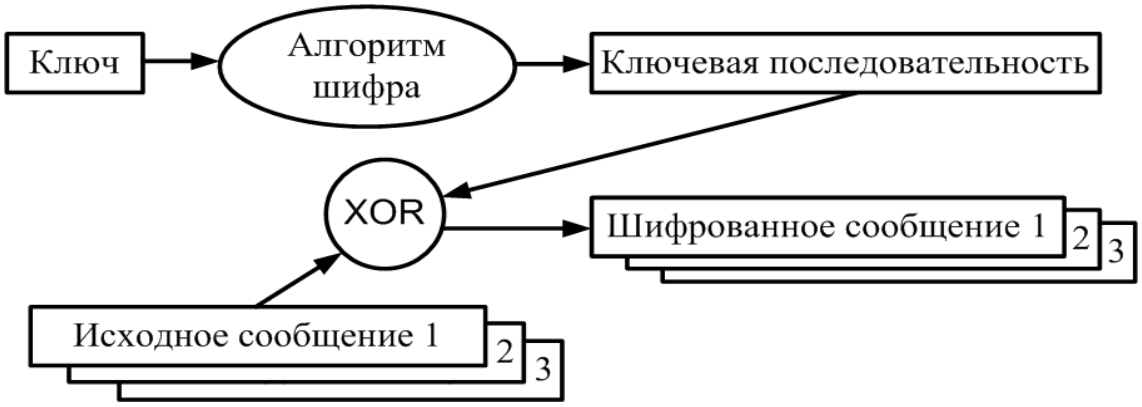


Рисунок 6.3 - Блочное шифрование.

Для устранения указанной проблемы используют:

− векторы инициализации (Initialization Vectors - IVs);

обратную связь (feedback modes).

До начала процесса шифрования 40- или 104-битный секретный ключ распределяется между всеми станциями, входящими в беспроводную сеть. К секретному ключу добавляется вектор инициализации (IV).

− Вектор инициализации (Initialization Vector, IV)

Вектор инициализации используется для модификации ключевой последовательности. При использовании вектора инициализации ключевая

последовательность генерируется алгоритмом шифрования, на вход которого подается секретный ключ, совмещенный с IV. При изменении вектора инициализации ключевая последовательность также меняется. На рисунке 5.4 исходное сообщение шифруется с использованием новой ключевой последовательности, сгенерированной алгоритмом шифрования после подачи на его вход комбинации из секретного ключа и вектора инициализации, что порождает на выходе шифрованное сообщение.

Стандарт IEEE 802.11 рекомендует использовать новое значение вектора инициализации для каждого нового фрейма, передаваемого в радиоканал.

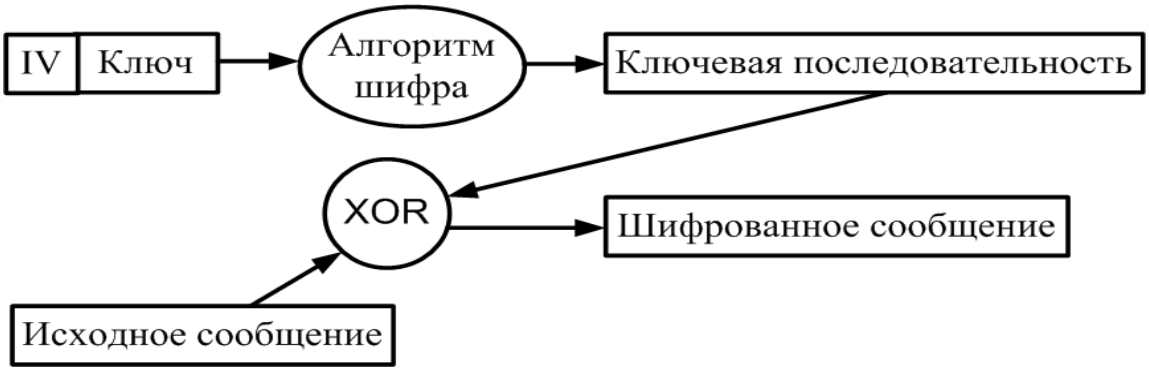


Рисунок 6.4 - Алгоритм шифрования WEP.

Новый стандарт безопасности WPA обеспечивает уровень безопасности куда больший, чем может предложить WEP Он перебрасывает мостик между стандартами WEP и 802.11 i и имеет немаловажное преимущество, которое заключается в том, что микропрограммное обеспечение более старого оборудования может быть заменено без внесения аппаратных изменений.

IEEE предложила временный протокол целостности ключа (Temporal Key Integrity Protocol, TKIP).

Основные усовершенствования, внесенные протоколом TKIP:

− покадровое изменение ключей шифрования. WEP-ключ быстро изменяется, и для каждого кадра он другой;

− контроль целостности сообщения. Обеспечивается эффективный контроль целостности фреймов данных с целью предотвращения скрытых манипуляций с фреймами и воспроизведения фреймов;

− усовершенствованный механизм управления ключами.

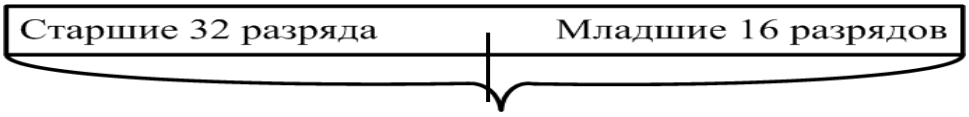
Пофреймовое изменение ключей шифрования

Атаки, применяемые в WEP, использующие уязвимость слабых IV (Initialization Vectors), таких, которые применяются в приложении AirSnort, основаны на накоплении нескольких фреймов данных, содержащих информацию, зашифрованную с использованием слабых IV. Простейшим способом сдерживания таких атак является изменение WEP-ключа, используемого при обмене фреймами между клиентом и точкой доступа, до того

как атакующий успеет накопить фреймы в количестве, достаточном для вывода битов ключа.

IEEE адаптировала схему, известную как пофреймовое изменение ключа (per-frame keying). Основной принцип, на котором основано пофреймовое изменение ключа, состоит в том, что IV, MAC-адрес передатчика и WEP-ключ обрабатываются вместе с помощью двухступенчатой функции перемешивания. Результат применения этой функции соответствует стандартному 104-разрядному WEP-ключу и 24-разрядному IV.

IEEE предложила также увеличить 24-разрядный вектор инициализации до 48-разрядного IV. На рисунке 5 представлен образец 48-разрядного IV и показано, как он разбивается на части для использования при пофреймовом изменении ключа.



48 разрядов IV

Рисунок 6.5 - Разбиение 48-разрядного IV.

Процесс пофреймового изменения ключа можно разбить на следующие этапы (рисунок 6):

− базовый WEP-ключ перемешивается со старшими 32 разрядами 48-разрядного IV (32-разрядные числа могут принимать значения 0-4 294 967 295) и MAC-адресом передатчика. Результат этого действия называется ключ 1-й фазы. Этот процесс позволяет занести ключ 1-й фазы в кэш и также напрямую поместить в ключ;

− ключ 1-й фазы снова перемешивается с IV и MAC-адресом передатчика для выработки значения пофреймового ключа;

− вектор инициализации (IV), используемый для передачи фрейма, имеет размер только 16 бит (16-разрядные числа могут принимать значения 0-65 535). Оставшиеся 8 бит (в стандартном 24-битовом IV) представляют собой фиксированное значение, используемое как заполнитель;

− пофреймовый ключ применяется для WEP-шифрования фрейма данных.

Когда 16-битовое пространство IV оказывается исчерпанным, ключ 1-й фазы отбрасывается и 32 старших разряда увеличиваются на 1. Значение пофреймового ключа вычисляется заново, как на этапе 2.

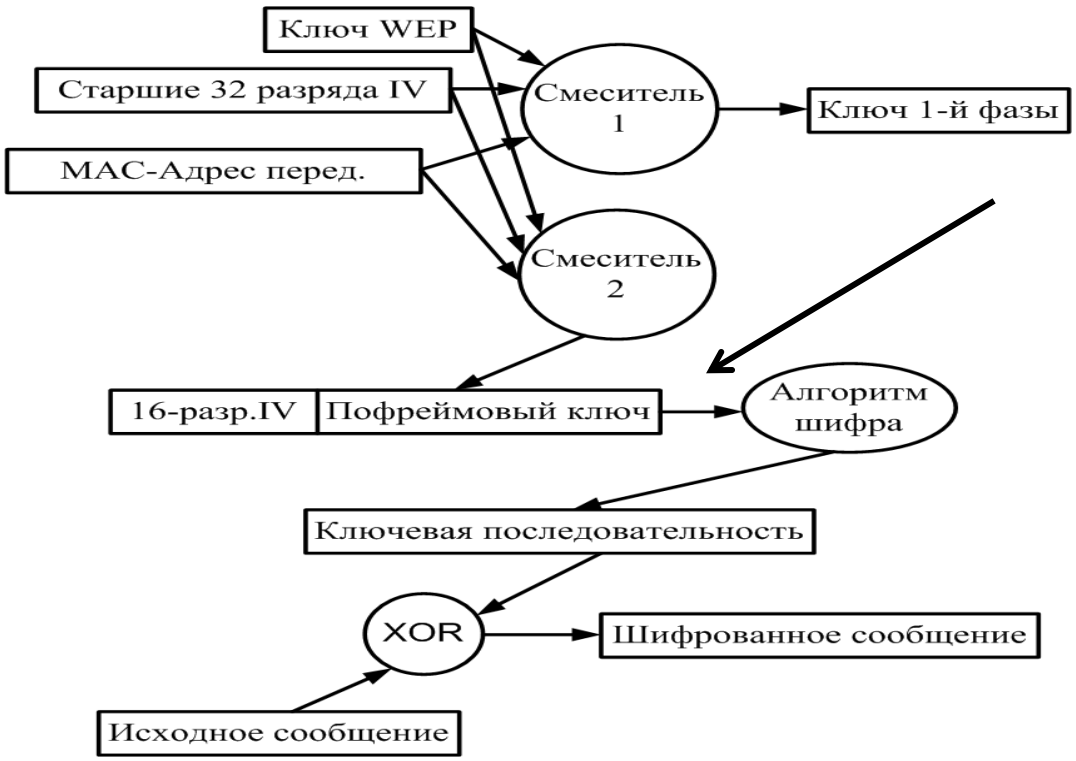


Рисунок 6.6. Процесс создания шифрованного сообщения в WPA.

Процесс пофреймового изменения ключа можно разбить на следующие этапы.

− устройство инициализирует IV, присваивая ему значение 0. В двоичном представлении это будет значение 00000000000000000000 0000000000 000000000000000000;

− первые 32 разряда IV (в рассматриваемом случае - первые 32 нуля) перемешиваются с WEP-ключом (например, имеющим 128-разрядное значение) и MAC-адресом передатчика (имеющим 48-разрядное значение) для получения значения ключа 1-й фазы (80-разрядное значение);

− ключ 1-й фазы вновь перемешивается с первыми (старшими) 32 разрядами IV и MAC- адресом передатчика, чтобы получить 128-разрядный пофреймовый ключ, первые 16 разрядов которого представляют собой значение IV (16 нулей);

− вектор инициализации пофреймового ключа увеличивается на 1. После того как пофреймовые возможности IV будут исчерпаны, IV 1-й фазы (32 бита) увеличивается на 1 (он теперь будет состоять из 31 нуля и одной единицы, 00000000000000000000000000000001)и т. д.

Меры противодействия MIC состоят в выполнении приемником следующих задач:

Приемник удаляет существующий ключ на ассоциирование.

Приемник регистрирует проблему как относящуюся к безопасности сети.

Ассоциированный клиент, от которого был получен ложный фрейм, не

может быть ассоциирован и аутентифицирован в течение 60 секунд, чтобы замедлить атаку.

Клиент запрашивает новый ключ.

WPA может работать в двух режимах: Enterprise (корпоративный) и Pre-Shared Key (персональный).

В первом случае хранение базы данных и проверка аутентичности по стандарту 802.1x в больших сетях обычно осуществляются специальным сервером, чаще всего RADIUS (Remote Authentication Dial-In User Service). Enterprise-режим мы рассмотрим далее.

Во втором случае подразумевается применение WPA всеми категориями пользователей беспроводных сетей, т.е. имеет место упрощенный режим, не требующий сложных механизмов. Этот режим называется WPA-PSK и предполагает введение одного пароля на каждый узел беспроводной сети (точку доступа, беспроводной маршрутизатор, клиентский адаптер, мост). До тех пор, пока пароли совпадают, клиенту будет разрешен доступ в сеть. Можно заметить, что подход с использованием пароля делает WPA-PSK уязвимым для атаки методом подбора, однако этот режим избавляет от путаницы с ключами WEP, заменяя их целостной и четкой системой на основе цифро-буквенного пароля.

Таким образом, WPA/TKIP - это решение, предоставляющее больший по сравнению с WEP уровень безопасности, направленное на устранение слабостей предшественника и обеспечивающее совместимость с более старым оборудованием сетей 802.11 без внесения аппаратных изменений в устройства.

3. Стандарт сети 802.11 i с повышенной безопасностью (WPA2)

В июне 2004 г. IEEE ратифицировал давно ожидаемый стандарт обеспечения безопасности в беспроводных локальных сетях - 802.11i.

Действительно, WPA достоин восхищения как шедевр ретроинжиниринга. Созданный с учетом слабых мест WEP, он представляет собой очень надежную систему безопасности и, как правило, обратно совместим с существующим Wi-Fi-оборудованием. WPA - практическое решение, обеспечивающее достаточный уровень безопасности для беспроводных сетей.

Однако WPA - компромиссное решение. Оно все еще основано на алгоритме шифрования RC4 и протоколе TKIP. Вероятность выявления каких-либо слабых мест хотя и мала, но все же существует.

Абсолютно новая система безопасности, лишенная недостатков WEP, представляет собой лучшее долгосрочное и к тому же расширяемое решение для безопасности беспроводных сетей. С этой целью комитет по стандартам принял решение разработать систему безопасности с нуля. Это новый стандарт 802.11i, также известный как WPA2 и выпущенный тем же Wi-Fi Alliance.

Стандарт 802.11i использует концепцию повышенной безопасности (Robust Security Network, RSN), предусматривающую, что беспроводные устройства должны обеспечивать дополнительные возможности. Это потребует изменений в аппаратной части и программном обеспечении, т.е. сеть, полностью соответствующая RSN, станет несовместимой с существующим оборудованием WEP. В переходный период будет поддерживаться как оборудование RSN, так и WEP (на самом деле WPA/TKIP было решением, направленным на сохранение инвестиций в оборудование), но в дальнейшем устройства WEP начнут отмирать.

802.11i приложим к различным сетевым реализациям и может задействовать TKIP, но по умолчанию RSN использует AES (Advanced Encryption Standard) и CCMP (Counter Mode CBC MAC Protocol) и, таким образом, является более мощным расширяемым решением.

В концепции RSN применяется AES в качестве системы шифрования, подобно тому как алгоритм RC4 задействован в WPA. Однако механизм шифрования куда более сложен и не страдает от проблем, свойственных WEP AES - блочный шифр, оперирующий блоками данных по 128 бит. CCMP, в свою очередь, - протокол безопасности, используемый AES. Он является эквивалентом TKIP в WPA. CCMP вычисляет MIC, прибегая к хорошо известному и проверенному методу Cipher Block Chaining Message Authentication Code (CBC-MAC). Изменение даже одного бита в сообщении приводит к совершенно другому результату.

Одной из слабых сторон WEP было управление секретными ключами. Многие администраторы больших сетей находили его неудобным. Ключи WEP не менялись длительное время (или никогда), что облегчало задачу злоумышленникам.

RSN определяет иерархию ключей с ограниченным сроком действия, сходную с TKIP В AES/CCMP, чтобы вместить все ключи, требуется 512 бит - меньше, чем в TKIP В обоих случаях мастер-ключи используются не прямо, а для вывода других ключей. К счастью, администратор должен обеспечить единственный мастер-ключ. Сообщения составляются из 128-битного блока данных, зашифрованного секретным ключом такой же длины (128 бит). Хотя процесс шифрования сложен, администратор опять-таки не должен вникать в нюансы вычислений. Конечным результатом является шифр, который гораздо сложнее, чем даже WPA.

IEEE 802.11i (WPA2) - это наиболее устойчивое, расширяемое и безопасное решение, предназначенное в первую очередь для крупных предприятий, где управление ключами и администрирование доставляет множество хлопот.

Преимущества WPA перед WEP

WPA, WiFi Protected Access – новый, самый современный, на данный момент, механизм защиты беспроводных сетей от неавторизованного доступа. WPA, и его дальнейшее развитие WPA2 пришло на замену механизму WEP, который начал устаревать к тому времени. Еще раз, рассмотрим принцип

действия WEP:

1. Кадр данных состоит из зашифрованной и незашифрованной части. Зашифрованная часть сдержит в себе данные и контрольную суммы(CRC32), незашифрованная – вектор инициализации и идентификатор ключа.

2. Каждый кадр данных шифруется поточным шифром RC4, используя в качестве ключа шифрования вектор инициализации с присоединенным к нему ключом WEP.

Таким образом, для каждого кадра данных генерируется свой ключ шифрования, однако в то же время каждый новый ключ шифрования отличается от другого всего лишь на вектор инициализации. (24 бита, когда длина ключа может быть 40 либо 104 бит). Таким образом, если злоумышленник перехватит большое количество пакетов, то он получит следующее: -большое количество векторов инициализации

-большое количество зашифрованных данных

-ключ шифрования для каждого последующего кадра отличается от предыдущего всего на 4 бита (длина вектора инициализации)

Таким образом, есть возможность извлечения ключа путем выполнения математических операций над пакетами.

Для того, чтобы успешно получить ключ WEP, злоумышленнику необходимо следующее:

-находиться в том месте, где есть возможность приема сигнала сети (вполне хватит RSSI в -85 dBm)

-захватить около 100-200 тыс. векторов инициализации, в зависимости от длины ключа (WEP-40 или WEP-104). Обычно, для этого нужно перехватить 25-50 Мб трафика, передающегося в сети. При наличии высокой сетевой активности (загрузка файлов (особенно с использованием пиринговых сетей) видеоконференции), для захвата необходимого объема трафика хватит 5-10 минут.

Также обратите внимание на то, как злоумышленник выполняет захват трафика. Обычно, беспроводные сетевые адаптеры работают в нормальном режиме – принимают только те пакеты, которые посланы на их MAC-адрес, при условии, что они подключены к этой беспроводной сети. Однако, физически ни что не мешает беспроводному сетевому адаптеру захватывать все пакеты, которые есть в радиусе его действия на выбранном канале. Для реализации такой возможности существуют специальные неофициальные драйвера и программное обеспечение. Больше того, такое программное обеспечение продается вполне легально, и оно используется для мониторинга беспроводных сетей. Примером такой программы может служить CommView for WiFi компании TamoSoft. Далее злоумышленник выполняет анализ захваченного трафика. Поскольку WEP был взломан уже много лет назад, то в интернете можно найти утилиты, которые в автоматическом режиме извлекают ключ из CAP-файла с трафиком, самой распространенной среди них является Aircrack.

Таким образом, WEP имеет следующие минусы

-предсказуемость ключа шифрования для кадра

-отсутствие средств аутентификации в сети

-слабый механизм проверки целостности данных

Поэтому многие предприятия отказывались от использования беспроводных сетей вообще, чтобы избежать утечку корпоративной информации. Однако с появлением WPA, а в последствии WPA2 ситуация изменилась, и все больше и больше корпоративных пользователей стали использовать WPA. Действительно, в сравнении с WEP он обладает рядом преимуществ:

-математическая независимость друг от друга ключей шифрования для каждого пакета

-новый механизм подсчета контрольной суммы

-WPA включает в себя средства аутентификации протокола 802.1X

Принцип работы WPA

Первые модификации WPA представляли собой усовершенствованный WEP.

Рассмотрим один из первых протоколов WPA, WPA-TKIP

В нем используется 48-битный вектор инициализации, и изменены правила построения вектора, также для подсчета контрольной суммы используется MIC (Message Integrity Code), который используется вместо устаревшего и менее надёжного CRC32

И самым главным усовершенствованием является то, что длина ключа шифрования теперь составляет 128 бит, вместо 40. Для управления ключами существует специальная иерархия, которая призвана предотвратить предсказуемость ключа шифрования для каждого кадра. Благодаря TKIP ключ шифрования для каждого кадра данных генерируется таким образом, что они не повторяют друг друга, пусть даже частично.

Таким образом, WPA-сети полностью защищены от атак replay (повторение ключей) и forgery (подмена содержимого пакетов), чего нельзя было сказать о WEP, где было возможно обойти CRC32-проверку контрольной суммы, а также отослать кадр с точно таким же ключом шифрования, как и у предыдущего.

Контрольные вопросы.

1. Используется ли WEP в данный момент;
2. Что такое потоковые шифрования;
3. Что такое блочное шифрования;
4. Что такое TKIP и как он создается;
5. Для чего нужен вектор инициализации.

Лекция 7. Виртуальные частные сети (Virtual Private Network, VPN)

Вопросы лекции:

1) Виртуальные частные сети.

2) Обзор протоколов PPTP и L2TP .
3) Протокол IPSEC.
4) Ассоциация обеспечения безопасности.
5) Протокол аутентификации заголовка.
6) Протокол SSL/TLS

Цель занятия ознакомить студентов виртуальными частными сетями методами их создания какие протоколы используется при этом и на каких уровня МВОС можно создать.

1. Виртуальные частные сети

Виртуальные частные сети, или защищенные виртуальные сети (Virtual Private Network, VPN), - это подключение, установленное по существующей общедоступной инфраструктуре и использующее шифрование и аутентификацию для обеспечения безопасности содержания передаваемых пакетов. Виртуальная частная сеть создает виртуальный сегмент между любыми двумя точками доступа к сети. Она может проходить через общедоступную инфраструктуру локальной вычислительной сети, подключения к глобальной сети (Wide area Network, WAN) или Интернет.

Рассмотрим VPN, организующие безопасные каналы передачи данных, использующие общедоступную инфраструктуру или Интернет. Все VPN по конфигурации можно подразделить на три основных типа:

− узел-узел (host-to-host);
− узел-шлюз (host-to-gateway);
− шлюз-шлюз (gateway-to-gateway).

Для организации канала связи, проходящего через Интернет, можно использовать VPN любого типа. Основной концепцией VPN является защита шифрованием канала связи на различных уровнях модели TCP/IP, а именно:

− прикладном (5-й уровень);
− транспортном (4-й уровень);
− сетевом (3-й уровень);
− канальном (2-й уровень).

Схема расположения протоколов VPN по уровням модели приведена в таблице 7.1.

На прикладном уровне шифрование можно применять при помощи программ, подобных пакету Pretty Good Privacy (PGP), или через каналы типа Secure Shell (SSH). Такие программы работают на участке сети от узла до узла, что означает, что они предлагают защиту только душ содержимого (payload) пакета, а не всего пакета в целом. Исключение составляет протокол SSH, который может использовать режим port-forwarding для создания туннеля.

Таблица 7.1 - Схема расположения протоколов VPN по уровням модели.

Уровни TCP/IP	Основные протоколы

Прикладной	PGP, S/MIME SSH, Kerberos, RADIUS
Транспортный	SSL, TSL SOCKS v5
Уровень межсетевого взаимодействия (сетевой)	IPSec (AH, ESP)
Уровень сетевых интерфейсов (канальный)	L2TP, PPTP

На транспортном уровне для защиты содержимого пакетов конкретного сеанса между двумя сторонами можно использовать протоколы, аналогичные протоколу защищенных сокетов (Secure Sockets Layer, SSL). Обычно такой метод используется при соединениях, установленных посредством Web-браузера. При этом также защищается только содержательная часть передаваемых пакетов, а IP-датаграммы, которые несут эту информацию, доступны для просмотра.

На сетевом уровне протоколы, подобные IPSec, не только зашифровывают содержательную часть пакета (полезную нагрузку), но и зашифровывают информацию заголовков протоколов TCP/IP.

На канальном уровне протокол туннелирования (Layer 2 Tunneling Protocol, L2TP) является расширением протокола соединения типа «точка-точка» (Point-to-Point Protocol, PPP), который допускает шифрование пакетов, посланных по PPP-протоколу на канальном уровне передачи данных.

Несмотря на то, что эти технологии шифрования применяются на разных уровнях, они все могут быть частью VPN. Необходимо отметить, что некоторые из этих технологий не могут обрабатывать все режимы работы VPN без дополнительной помощи со стороны других приложений или протоколов.

2. Обзор протоколов PPTP и L2TP

На канальном уровне существуют два протокола для реализации VPN: протокол туннелирования типа «точка-точка» (Point-to-point Tunneling Protocol, PPTP) и протокол туннелирования второго уровня (Layer Two Tunneling Protocol, L2TP).

Протокол PPTP

Протокол PPTP является дальнейшим развитием протокола PPP, который распространился в связи с появлением модемного доступа к сети Интернет. Протокол PPTP был одним из первых протоколов для создания VPN через коммутируемое соединение и был разработан консорциумом таких производителей, как Microsoft, US Robotics, Ascend и 3Com. Для шифрования протокола PPP был использован протокол двухточечного шифрования Microsoft (Microsoft Point-to-Point Encryption, MPPE), который использует алгоритм RC4. Однако большинство проблем в области безопасности было связано с ненадежностью используемого метода аутентификации — протокола Microsoft аутентификации с предварительным согласованием вызова (Microsoft Challenge/Reply HandShake Protocol, MSCHAP). Для устранения недостатков был выпущен протокол MSCHAP версии 2. Протокол PPTP использует все связанные протоколы, которые подобны протоколу

MSCHAP, протоколу аутентификации пароля (Password Authentication Protocol, PAP), протоколу аутентификации с предварительным согласованием вызова (Challenge Handshake Authentication Protocol, CHAP), расширенному протоколу аутентификации (Extensible Authentication Protocol, EAP).

Протокол PPTP использует два канала, работающих совместно. Первый — канал управления (порт 1723/tcp). Этот канал посылает в обе стороны все команды, которые управляют сеансом подключения. Второй — инкапсулированный канал передачи данных, являющийся вариантом протокола общей инкапсуляции для маршрутизации (Generic Routing Encapsulation, GRE, IP-протокол 47). Преимуществом туннеля протокола общей инкапсуляции для маршрутизации является то, что он может инкапсулировать и передавать протоколы, отличающиеся от протокола IP. Первоначально клиент устанавливает соединение с провайдером с помощью PPP, а затем устанавливает TCP/IP- соединение с PPTP-сервером через интернет.

В отличие от L2TP PPTP не требует развёртывания инфраструктуры для сертификатов, а использует парольную аутентификацию, что упрощает развёртывание VPN, но в некоторых случаях снижает безопасность. PPTP-сервер имеет ограничение по количеству одновременных подключений. PPTP не использует IPSec, поэтому проблем с трансляцией адресов не возникает (не требуется механизм NAT Traversal).

Протокол PPTP реализован во многих аппаратных устройствах. Протокол PPTP при инициализации связи использует протокол PPP, по этому может оказаться уязвимым к атакам типа spoofing и «человек посередине».

Протокол L2TP (Layer 2 Tunneling Protocol) - открытый стандарт IETF, который решает многие проблемы PPTP, и определен в документе RFC 2661. Протокол L2TP фактически является гибридом двух предыдущих протоколов туннелирования: протокола пересылки второго уровня (Layer Two Forwarding, L2F) компании Cisco и протокола PPTP.

Протокол L2TP, подобно протоколу PPTP, использует при аутентификации пользователя возможности протокола PPP (протоколы MSCHAP, CHAP, EAP, PAP и т.д.). Аналогично протоколу PPTP протокол L2TP использует два канала связи: сообщения управления и сообщения туннеля для передачи данных. Первый бит заголовка протокола PPTP служит для опознания этих типов сообщений (1 — для сообщений управления, 0 — для сообщений данных). Сообщениям управления дается более высокий приоритет по отношению к сообщениям данных, чтобы гарантировать, что важная информация администрирования сеанса будет передана настолько быстро, насколько это возможно.

Подключение канала управления устанавливается для туннеля, который затем сопровождается инициированием сеанса протокола L2TP. После завершения инициирования обоих подключений информация в виде кадров протокола PPP начинает передаваться по туннелю.

Формирование защищенного канала происходит в три этапа:

− Установление соединения клиента с сервером удаленного доступа.
− Аутентификация пользователя.
− конфигурирование защищенного туннеля.

Для установления соединения с сервером удаленного доступа (сетевой сервер L2TP) удаленный пользователь связывается по протоколу PPP с концентратором доступа L2TP (Local Access Concentrator, LAC), обычно функционирующем на сервере провайдера. Концентратор доступа может выполнить аутентификацию пользователя от имени провайдера. По заданному имени получателя концентратор доступа определяет адрес сетевого сервера L2TP (L2TP Network Server, LNS), который защищает сеть с заданным адресом. Между концентратором доступа и сервером L2TP устанавливается соединение. Далее производится аутентификация пользователя сервером L2TP. В случае успешной аутентификации устанавливается защищенный туннель между концентратором доступа и сервером L2TP. С помощью управляющих сообщений производится настройка параметров туннеля, причем в одном туннеле может быть несколько сеансов пользователя. При использовании IPSec пакеты L2TP инкапсулируются в UDP-па-кеты, которые передаются концентратором доступа и сервером L2TP через IPSec-туннель (порт 1701/tcp). В большинстве случаев клиент сам выступает в роли LAC.

L2TP использует сертификаты для аутентификации, за счёт чего упрощается администрирование сервера с большим количеством клиентов и повышается безопасность. В отличие от PPTP имеется возможность создания нескольких виртуальных сетей через один туннель. В случае, когда L2TP использует IPSec, при необходимости трансляции адресов требуется NAT Traversal на LNS.

Сравнение PPTP/L2TP

В PPTP шифрование данных начинается после установления PPP-соединения. В L2TP/IPSec шифрование начинается до установления такого соединения.

В PPTP для шифрования применяется протокол MPPE, основанный на алгоритме RSA RC4, который обеспечивает возможность применения 40-, 56- и 128-битных ключей. В L2TP/IPSec возможно применение любых алгоритмов шифрования, поддерживаемых IPSec.

В PPTP используется аутентификация PPP (сервер аутентифицирует клиента по паре логин-пароль). В L2TP/IPSec наряду с учётной парой применяется и аутентификация по сертификатам.

3. Протокол IPSEC

Несмотря на то, что протокол IP стал наиболее используемым протоколом связи во всем мире и базовой технологией Интернета, он имеет множество существенных недостатков. Среди них необходимо отметить ограниченность адресного пространства и недостатки внутренней безопасности. Главной причиной этих недостатков является то, что IP-

протокол изначально не был предназначен для массового использования.

В качестве развития IP-протокола была разработана его новая версия IPv6 (IP-протокол версии 6), в которой пытались решить проблемы предшествующих версий. Поскольку принятие новой версии IP-протокола затруднительно, что вызвано постоянным ростом Интернета и громадным разнообразием установленной аппаратуры и программного обеспечения, то меры безопасности, включенные в IPv6, были перенесены в текущую версию IPv4 в виде дополнительного комплекта протоколов. Этот набор протоколов назвали комплектом протоколов IPSec (IPSec Protocol Suite).

Подключение по протоколу IPSec имеет два основных режима: транспортный (transport) и туннельный (tunnel).Транспортный режим — это форма связи типа узел-узел, где применяется шифрование только содержательной части пакета. Из-за двухточечного характера связи соответствующее программное обеспечение необходимо загрузить (установить) на все связывающиеся между собой узлы сети, что представляет собой достаточно серьезную проблему. Этот режим VPN удобно использовать для зашифрованной связи между узлами одной сети.

Режим туннелирования применяется при создании большинства VPN; потому что он шифрует весь оригинальный пакет. Режим туннелирования может применяться для организации связи типа узел-узел, узел-шлюз или шлюз-шлюз. При организации связи типа шлюз-шлюз значительно упрощается связь между сетями, не требуется установка специального ПО на узлах сети.

Первая цель семейства протоколов IPSec состоит в обеспечении конфиденциальности, целостности и аутентификации информации, передаваемой посредством IP-протокола. Это достигается с помощью протокола обмена интернет-ключами (Internet Key Exchange, IKE), протокола ESP и протокола AH. Комбинация этих трех протоколов обеспечивает безопасный обмен информацией.

Второй целью семейства протоколов IPSec является предоставление разработчикам ПО набора стандартов. Установление безопасного соединения начинается с формирования ассоциации обеспечения безопасности (Security Association, SA) между двумя сообщающимися сторонами.

4. Ассоциация обеспечения безопасности.

Основа ассоциации обеспечения безопасности заключается в соглашении двух сторон о том, как они могут безопасно передавать свою информацию. В процессе соглашения стороны оговаривают детали защищенного обмена. Результатом такого соглашения и является ассоциация обеспечения безопасности. Каждому сеансу связи сопоставляются две ассоциации — по одной на каждого партнера связи.

Положительными чертами протокола IPSec являются открытость его стандарта для поддержки множества протоколов и режимов связи, а также

поддержка различных алгоритмов шифрования и различных хэш-функций. Прежде чем договариваться об ассоциации обеспечения безопасности, необходимо локальное конфигурирование элементов протокола IPSec, которые данный партнер собирается поддерживать. Эти параметры настройки хранятся в базе данных политики безопасности (Security Policy Database, SPD).

После согласования ассоциация обеспечения безопасности содержится в базе данных ассоциации обеспечения безопасности (Security Association Database, SAD). Это необходимо, поскольку узел сети может инициализировать несколько сеансов, каждому из которых может соответствовать своя SA.

Поскольку для каждого сетевого устройства доступно множество сеансов протокола IPSec, для правильного функционирования процесса необходимо, чтобы каждый сеанс согласования SA имел свой собственный уникальный идентификатор. Этот идентификатор составляется из уникального индекса параметра обеспечения безопасности (Security Parameter Index, SPI), который определяет, какая запись БД SA соответствует рассматриваемому подключению. Кроме того, учитывается адрес назначения и идентификатор используемого протокола (ESP или AH).

Протокол обмена интернет-ключами

Протокол обмена интернет-ключами предназначен для аутентификации и согласования параметров обмена протокола IPSec. Протокол IKE представляет собой комбинацию двух протоколов: протокола управления ассоциациями и протокола управления ключами обеспечения безопасности в сети Интернет (Internet Security Association and Key Management Protocol, ISAKMP), называемых фазами установления. Управление ключами можно выполнять вручную или используя альтернативы протокола IKE, такие как безопасная служба доменных имен (Secure DNS), Photuris или простой протокол обмена интернет- ключами (Simple Key Internet Protocol, SKIP).

Первая фаза протокола IKE. На первой фазе протокола IKE удаленный пользователь начинает сеанс со шлюзовым устройством VPN. Первая фаза выполняет две функции: аутентификацию удаленного пользователя и обмен информацией об открытых ключах, которые будут использоваться во второй фазе.

Аутентификацию можно выполнить несколькими различными способами. Наиболее часто используются технология предварительно распространяемых ключей (pre-shared keys) и технология цифровых сертификатов. Термин «предварительно распространяемые ключи» означает, что значения ключей предварительно задаются на всех компьютерах, которые собираются устанавливать соединения через VPN (что является существенным недостатком). При втором способе используются цифровые удостоверения (цифровые сертификаты, digital certificates), которые могут назначаться отдельно для каждого объекта, который соединяется с VPN. Цифровыми сертификатами можно удаленно управлять и администрировать

из уполномоченного центра сертификации (Certificate Authority, CA).

Сертификационная служба является центральным элементом структуры, называемой инфраструктурой с открытым ключом (Public Key Infrastructure, PKI). За инфраструктурой РК! стоит концепция публично доступной структуры, распределяющей информацию об открытых (публичных) ключах.

В первой фазе при обмене аутентификационной информацией и параметрами безопасности могут использоваться два режима: основной (main mode) и агрессивный (aggressive mode). Различия между ними заключаются в количестве сетевых пакетов, которыми обмениваются стороны, и во времени, за которое генерируется открытый ключ. Агрессивный режим использует дополнительный заголовок меньшего размера, но основной режим обладает большей безопасностью и используется чаще всего.

Вторая фаза протокола IKE. Во второй фазе протокола IKE согласовываются конкретные параметры ассоциации обеспечения безопасности IPSec. Данное согласование подобно агрессивному режиму обмена информацией первой фазы. После завершения второй фазы формируется SA и пользователь получает подключение к VPN.

Во второй фазе возможен единственный режим согласования - быстрый режим (quick mode). Быстрый режим представляет собой короткий обмен, использующий три пакета. Все обмены второй фазы зашифрованы с помощью согласованных во время первой фазы протоколов и типов кодирования. При этом используется только защита, основанная на использовании хэш-функции и нонсе (nonce), включаемых в сетевые пакеты для подтверждения их оригинальности (нонсе является подтверждением того факта, что информация о ключе исходит из ожидаемого источника. Нонсе - это случайное число, генерируемое инициатором связи, которое заверяется респондентом цифровой подписью и посылается обратно).

5. Протокол аутентификации заголовка

Протокол AH - это IP-протокол 51. Он поддерживает функциональные возможности аутентификации и проверки целостности, но не поддерживает конфиденциальность содержимого пакета.

Обеспечение аутентификации и защиты целостности достигается добавлением дополнительного заголовка к IP-пакету. Этот заголовок содержит цифровую подпись, называемую значением проверки целостности (Integrity Check Value, ICV), которая является в основном значением хэш-функции, подтверждающей, что пакет не был изменен во время транспорта.

Протокол AH также поддерживает использование порядковых номеров (sequence numbers), помогающих предотвращать нападения, основанные на повторном использовании пакета (replay attack). Эти номера используются коммуникационными устройствами для отслеживания потока сетевых пакетов сеанса связи.

Таблица 7.1 – IP заголовок

Следующий заголовок	Длина содержимого пакета	Зарезервировано
Индекс параметра обеспечения безопасности (SPI)		
Порядковый номер		
Информация аутентификации (переменная длина, кратная 32 байтам)		

Использование информации IP-заголовка протоколом AH делает его несовместимым с использованием NAT. Структура заголовка AH-пакета представлена на рисунке 4.

Поле следующего заголовка содержит идентификатор, определяющий тип заголовка пакета, следующего за заголовком AH-пакета. Поле длины содержимого пакета определяет длину заголовка AH-пакета. Индекс параметра обеспечения безопасности показывает, частью какого уникального потока связи ассоциации обеспечения безопасности является рассматриваемый пакет.

Аутентифицируется за исключением изменяемых полей

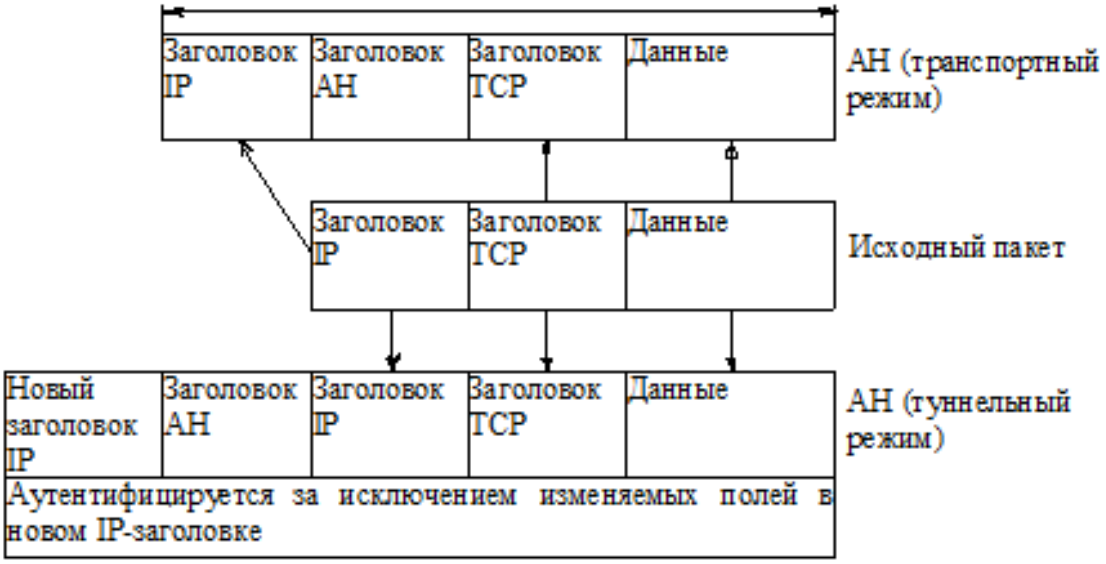


Рисунок 7.1 - Схема преобразования исходного пакета в AH-пакет.

Поле информации аутентификации содержит значение проверки целостности и цифровую подпись, подтверждающую подлинность рассматриваемого пакета. Схема преобразования исходного пакета в AH-пакет для различных режимов приведена на рисунке 5.

Протокол безопасной инкапсуляции содержимого пакета

Протокол ESP - это IP-протокол 50. Он обеспечивает конфиденциальность при помощи полного шифрования содержимого IP-пакетов. Протокол ESP реализован в виде модулей и может использовать любое количество доступных симметричных алгоритмов шифрования, в числе которых DES, 3DES, IDEA. Применение протокола ESP различается в зависимости от используемого режима протокола IPSec.

В транспортном режиме протокол ESP просто добавляет свой собственный заголовок после IP-заголовка и зашифровывает остальную часть сетевого пакета начиная с транспортного уровня. Если при этом определена служба аутентификации, то протокол ESP добавляет концевую метку (trailer). Концевая метка предназначена для подтверждения целостности пакета и аутентификации (в отличие от протокола AH значение проверки целостности вычисляется без использования информации из IP-заголовка).

При использовании туннельного режима протокол ESP инкапсулирует оригинальный пакет полностью, шифруя его целиком и создавая новый IP-заголовок и ESP-заголовок в устройстве туннелирования. Концевая метка также добавляется в случае выбора аутентификационного сервиса протокола ESP.

Поле длины заполнения указывает, насколько заполнено содержимое пакета (если такое вообще имеет место), чтобы длина содержимого пакета и последующие поля заголовка соответствовали требованию выравнивания длины сетевого пакета.

Поле следующего заголовка сообщает номер протокола пакета, который инкапсулирован внутри пакета протокола ESP. Поле информации аутентификации содержит дополнительное значение проверки целостности, которое доступно для пакетов протокола ESP.

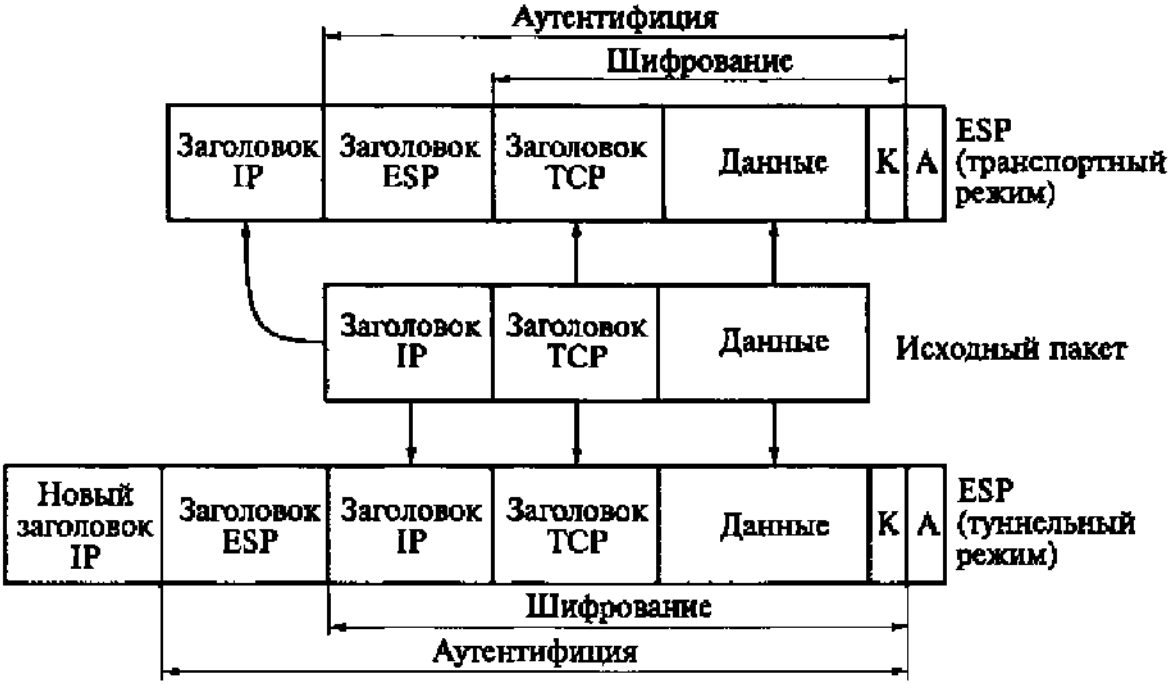


Рисунок 7.2 - Схема преобразования исходного пакета в пакет ESP.

Схема преобразования исходного пакета в пакет ESP для различных режимов приведена на рисунке 7.

Совместное использование протоколов ESP и AH

Отдельная защищенная связь может использовать либо протокол AH,

либо протокол ESP, но никак не оба эти протокола одновременно. Тем не менее, иногда конкретному потоку данных может требоваться и сервис AH, и сервис ESP. Во всех случаях одному потоку для получения комплекса услуг IPSec требуется несколько защищенных связей. Такой набор защищенных связей называется пучком защищенных связей (security association bundle), посредством которого потоку должен предоставляться необходимый набор услуг IPSec. При этом защищенные связи в пучке могут завершаться в различных конечных точках.

В случае применения ESP с опцией аутентификации пользователь сначала применяет ESP к требующим защиты данным, а затем добавляет поле данных аутентификации. В зависимости от используемых режимов возможны следующие варианты (в обоих вариантах аутентификации подлежит шифрованный текст):

транспортный режим ESP - аутентификация и шифрование применяются к полезному грузу IP, доставляемому узлу адресата, но при этом заголовок IP не защищается;

туннельный режим ESP - аутентификация применяется ко всему пакету IP, доставляемому по адресу IP внешнего получателя (например, МЭ), и выполняется этим получателем. Весь внутренний пакет IP защищается механизмом секретности, поскольку предназначен для доставки внутреннему адресату IP.

6. Протокол SSL/TLS

SSL (Secure Sockets Layer, уровень защищённых сокетов) - криптографический протокол, обеспечивающий безопасную передачу данных по сети Интернет. При его использовании создаётся защищённое соединение между клиентом и сервером. SSL изначально разработан компанией Netscape Communications. Впоследствии на основании протокола SSL 3.0 был разработан и принят стандарт RFC, получивший имя TLS. TLS является протоколом более низкого уровня, хотя на модели TCP/IP это не отражается (и SSL, и TLS - протоколы прикладного уровня).

Использует шифрование с открытым ключом для подтверждения подлинности передатчика и получателя. Поддерживает надёжность передачи данных за счёт использования корректирующих кодов и безопасных хэш-функций. Обычно применяются следующие алгоритмы:

Для обмена ключами: RSA, Diffie-Hellman, ECDH, SRP, PSK.

Для аутентификации: RSA, DSA, ECDSA.

Симметричные шифры: RC4, Triple DES, AES, IDEA, DES, или Camellia. В старых версиях SSL также использовался RC2.

Для криптографических хэшей: HMAC-MD5 или HMAC-SHA для TLS, MD5 и SHA для SSL.

В старых версиях SSL также использовались MD2 и MD4.

SSL состоит из двух уровней. На нижнем уровне многоуровневого

транспортного протокола (например, TCP) он является протоколом записи и используется для инкапсуляции (то есть формирования пакета) различных протоколов (SSL работает совместно с таким протоколами как POP3, IMAP, XMPP, SMTP и HTTP). Для каждого инкапсулированного протокола он обеспечивает условия, при которых сервер и клиент могут подтверждать друг другу свою подлинность, выполнять алгоритмы шифрования и производить обмен криптографическими ключами, прежде чем протокол прикладной программы начнёт передавать и получать данные. Если некоторая программа не поддерживает SSL/TLS, можно воспользоваться программой для создания защищенных туннелей stunnel.

Для обеспечения безопасности применяются следующие процедуры:

Клиент может использовать публичный ключ центра сертификации для проверки подлинности цифровой подписи серверного сертификата.

Клиент проверяет, что центр сертификации, подписавший ключ, относится к списку доверенных центров сертификации.

Клиент проверяет срок действия серверного сертификата.

Имеется защита от понижения версии протокола и от слабого алгоритма шифрования

Сообщение, завершающее "рукопожатие" (согласование), содержит хэш всех сообщений о согласовании, полученных обоими сторонами.

Псевдослучайная функция делит входные данные пополам и обрабатывает каждую часть своим алгоритмом хэширования (MD5 и SHA-1), а затем складывает их по модулю 2 (XOR), чтобы создать MAC (message authentification codes, коды аутентификации сообщения). Это обеспечивает защиту даже в том случае, если в одном из алгоритмов обнаружат уязвимость (только для TLS)

SSLv3 отличается от SSLv2 добавлением SHA-1 и поддержкой аутентификации сертификата. Также была улучшена процедура согласования и усилена устойчивость к атаке «человек посередине».

Область применения протоколов SSL/TLS весьма широка:

создание шифрованных туннелей;

шифрование текстовых протоколов интернета HTTP, POP3, IMAP, SMTP, XMPP;

аутентификация на интернет-сервисах или в локальной сети

Процедура установления шифрованной связи выглядит следующим образом:

Сервер и клиент, использующие TLS, устанавливают надёжное соединение с помощью процедуры согласования.

Согласование начинается, когда клиент подключается к использующему TLS серверу, запрашивая безопасное соединение, и представляет собой список поддерживаемых шифров и хэшей.

Из этого списка сервер выбирает наиболее мощные шифр и хэш и уведомляет клиента о решении.

Сервер отсылает свою идентификацию в виде цифрового сертификата.

Сертификат обычно содержит имя сервера, доверенный центр сертификации и публичный ключ шифрования сервера.

Перед продолжением процедуры клиент может связаться с центром сертификации, выдавшим сертификат, чтобы убедиться, что сертификат действителен:

Чтобы сгенерировать сеансовый ключ, используемый для защиты соединения, клиент шифрует случайное число с помощью публичного ключа сервера и посылает его серверу. Только сервер может расшифровать его (с помощью своего закрытого ключа). Это позволяет сделать ключ недосягаемым для третьих лиц.

Из этого случайного числа обе стороны генерируют ключ для шифрования и расшифровки.

Этот шаг завершает согласование и начинает защищенное соединение, передающиеся данные шифруются полученным ключом до закрытия соединения.

Если один из описанных шагов не был выполнен, то согласование прекращается и соединение не устанавливается.

Структура TLS-сообщения:

Content Type - 1 байт (тип содержимого);

Version (MSB) - 1 байт (версия протокола);

Version (LSB) - 1 байт (версия протокола);

Length (MSB) - 1 байт (длина сообщения);

Length (LSB) - 1 байт (длина сообщения);

Protocol Message(s) (сообщения протокола);

MAC (коды аутентификации сообщения, необязательно);

Padding (сдвиг, для блоковых шифров).

Последние 3 поля имеют переменную длину.

Контрольные вопросы.

1. Что за протокол SSL/TLS и для чего используется;
2. В чем разница протокола ESP от протокола AH;
3. В чем разница протокола PPTP от протокола L2TP;
4. Что такое TKIP и как он создается;
5. Как осуществляется ассоциация обеспечения безопасности.

Лекция 8. Безопасность удалённого управления

Вопросы лекции:

1) Аудит безопасности протокола SNMP.
2) Модель протокола SNMP Общая модель.
3) Формат SNMP-сообщения.
4) Протокол SSH.

Цель занятия ознакомить студентов удаленного управления сетевыми устройствами а также обеспечения безопасностью удаленного управления с

помощью протокола SSH.

1. Аудит безопасности протокола SNMP

Определение и функции протокола

Протокол SNMP (Simple Network Management Protocol, простой протокол управления сетью) является протоколом Прикладного уровня, разработанный для выполнения двух задач

мониторинг сетевых устройств и сети в целом;

управление сетевыми устройствами.

Протокол SNMP предоставляет возможность станциям управления считывать и изменять настройки шлюзов, маршрутизаторов, коммутаторов и прочих сетевых устройств.

Версии протокола SNMP

Опишем различия между версиями протокола SNMP и документы, определяющие эти версии. По состоянию на 2006 год единственной не устаревшей версией SNMP является SNMPv3, определённая в RFC 3411-3418.

SNMPv1

Первые RFC, описывающие стандарты SNMP, появились в 1988 году. Версия 1 подверглась критике за её посредственную модель безопасности на основе сообществ. В то время безопасность в Интернете не входила в круг первоочередных задач рабочих групп IETF.

SNMPv2

Версия 2, известная так же, как Party-based SNMPv2, или SNMPv2p, не получила широкого распространения из-за серьёзных разногласий по поводу инфраструктуры безопасности в стандарте. SNMPv2 улучшал версию 1 в области быстродействия, безопасности, конфиденциальности и взаимодействий «менеджер-менеждер». Он представил новый тип PDU Get-Bulk-Request, альтернативу Get-Next-Request для получения больших объёмов информации при помощи одного запроса. Тем не менее, новая система безопасности на основе сторон выглядела для многих как чересчур сложная и не была широко признана.

SNMPv3

SNMPv3 наконец-то решил проблемы с безопасностью способом, который многие посчитали приемлемым. Версия 3 SNMP принята IETF как стандарт Интернета (IETF STD 62). Почти все предыдущие RFC признаны устаревшими. Документы, описывающие протокол SNMPv3, приведены ниже:

2. Модель протокола SNMP Общая модель

Модель приведена на рисунке 1. Основными взаимодействующими элементами протокола являются агенты (agent) и системы управления сетью (NMS, network management system). С точки зрения концепции «клиент-сервер» роль сервера выполняют агенты, то есть те самые устройства, для опроса состояния, которых используется протокол SNMP. Соответственно,

роль клиентов отводится системам управления - сетевым приложениям, необходимым для сбора информации о функционировании агентов. Взаимодействие агентов и систем управления осуществляется на основе сообщений протокола SNMP.

Агентами в SNMP являются программные модули, которые работают в управляемых устройствах. Агенты собирают информацию об управляемых устройствах, в которых они работают. Агент содержат всю информация об управляемом сетевом устройстве в базе управляющей информации (MIB, management information base). MIB представляет собой совокупность объектов, доступных для операций записи-чтения.

В любой управляемой сети может иметься одна или более NMS. NMS выполняют прикладные программы сетевого управления, которые представляют информацию управления конечному пользователю.

Структура базы MIB

На данный момент существует четыре типа информационной базы MIB:

Internet MIB - информационная база объектов для обеспечения диагностики ошибок и конфигураций. Включает в себя 171 объект (в том числе и объекты MIB I).

LAN manager MIB - база из 90 объектов - пароли, сессии, пользователи, общие ресурсы.

WINS MIB - база объектов, необходимых для управления и диагностики WINS- сервера (в серверах Microsoft Windows физически находится в файле WINSMIB.DLL).

DHCP MIB - база объектов, необходимых для управления и диагностики DHCP- сервера (в серверах Microsoft Windows физически находится в файле DHCPMIB.DLL).

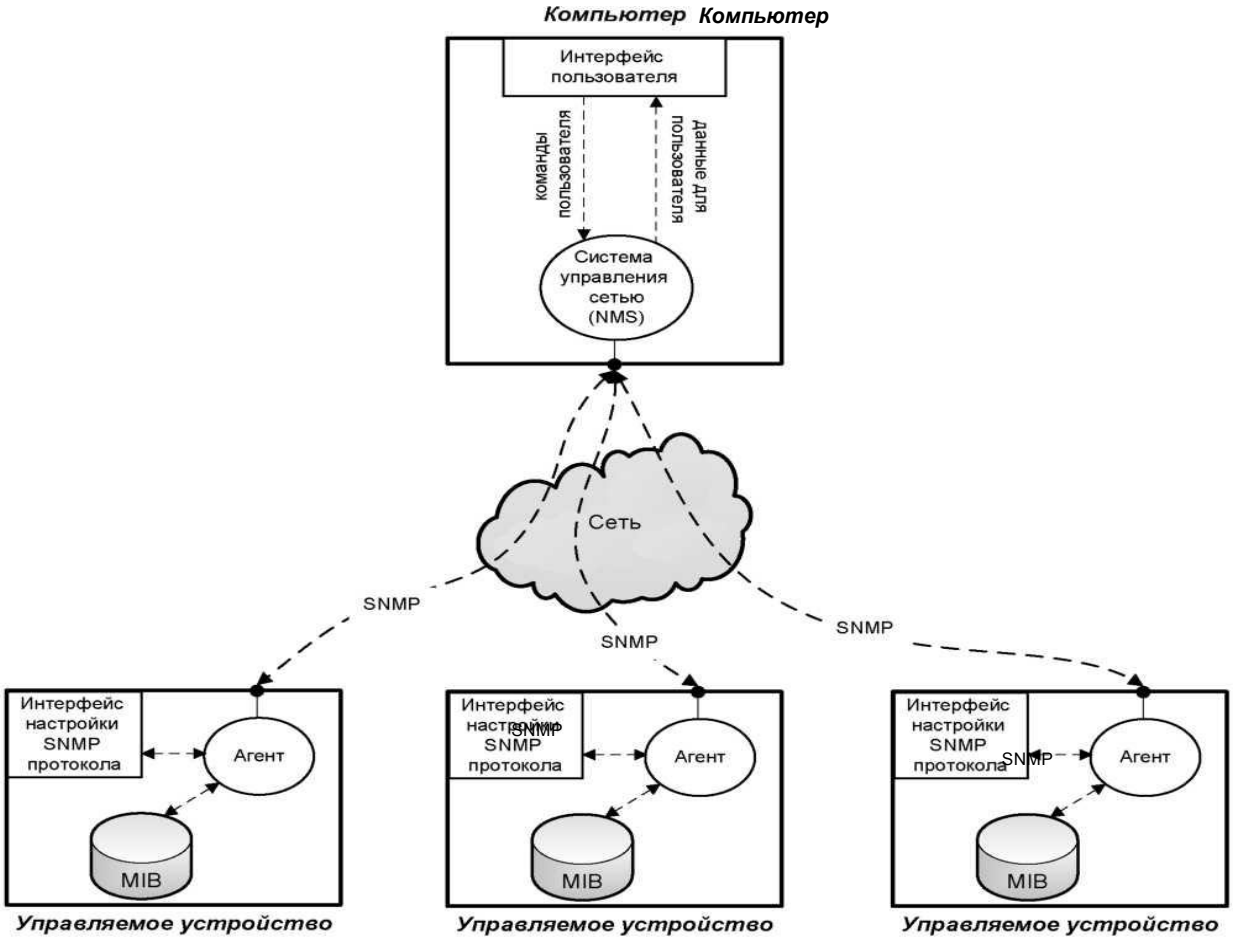


Рисунок 8.1 – Структура базы MIB

Структуру MIB определяет документ, называемый SMI (Structure of Management Information, структура управляющей информации). Все MIB имеют иерархическую древовидную структуру. Все базы содержат десять корневых алиасов (ветвей), представленных на рисунке 2:

System - данная группа MIB II содержит в себе семь объектов, каждый из которых служит для хранения информации о системе (версия ОС, время работы и т.д.).

Interfaces - содержит 23 объекта, необходимых для ведения агентами статистики по сетевым интерфейсам управляемого устройства (количество интерфейсов, размер MTU, скорость передачи данных, физические адреса и т.д.).

AT - содержит 3 объекта, отвечающих за трансляцию адресов. Более не используется. Была включена в MIB I. Примером использования объектов AT может послужить простая ARP таблица соответствия физических (MAC) адресов сетевых карт IP адресам машин. В SNMP v2 эта информация была перенесена в MIB для соответствующих протоколов.

IP - содержит 42 объекта, в которых хранятся данные о проходящих IP пакетах.

ICMP- содержит 26 объектов со статистикой об ICMP-сообщениях.

TCP - содержит 19 объектов, хранящих статистику по протоколу TCP (соединения, открытые порты и т.д.).

UDP - содержит 6 объектов, хранящих статистику по протоколу UDP (входящие/исходящие датаграммы, порты, ошибки).

EGP - содержит 20 объектов - данные о трафике Exterior Gateway Protocol.

Transmission - зарезервирована для специфических задач.

SNMP - содержит 29 объектов, в которых хранится статистика по SNMP-протоколу (входящие/исходящие пакеты, ограничения пакетов по размеру, ошибки, данные обобработанных запросах и многое другое).

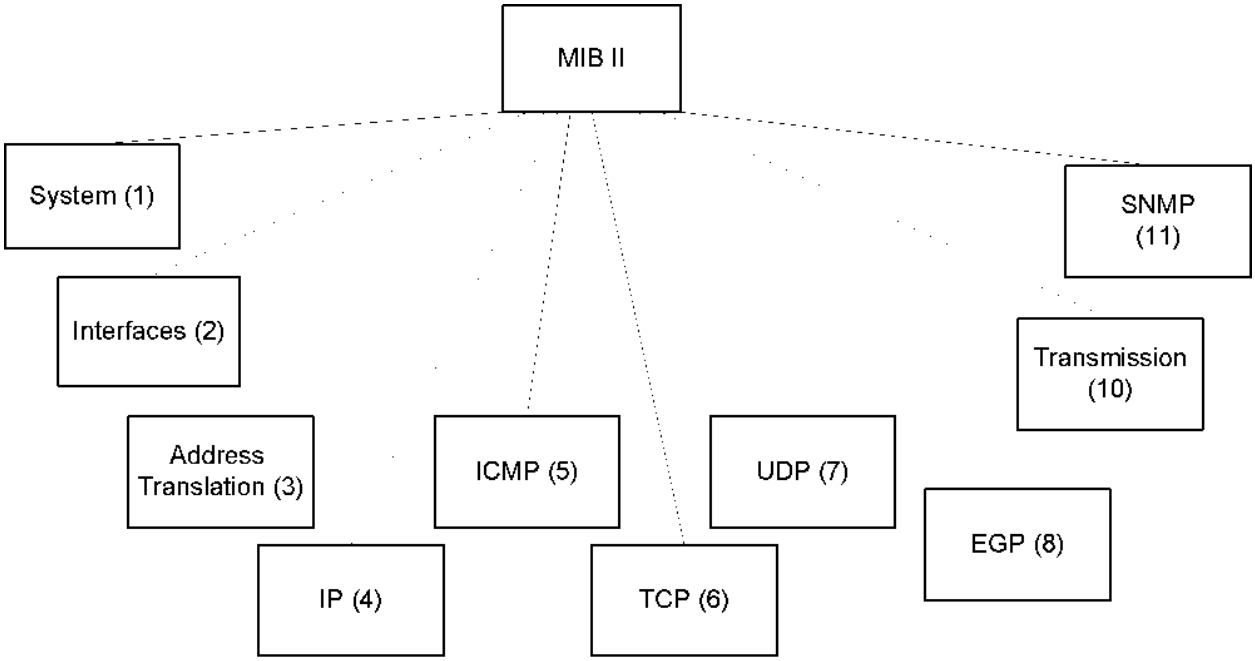


Рисунок 8.2 – MIB база

Дерево MIB расширяемо благодаря экспериментальным и частным ветвям. Например, поставщики могут определять свои собственные ветви для включения реализаций своих изделий. В настоящее время вся работа по стандартизации ведется на экспериментальной ветви.

Управляемые устройства используют «ловушки» для асинхронных сообщений в NMS о некоторых событиях.

Протокол SNMPv3

Начиная с января 1998 года, выпущен набор документов, посвященных SNMPv3. В этой версии существенно расширена функциональность, разработана новая система безопасности.

3. Протокол обмена данными

Ниже на рисунке 3 приведена временная диаграмма, на которой в общем виде представлен протокол обмена SNMP-сообщениями. Для своей работы

протокол SNMP использует транспортный протокол UDP, в основном 161 порт. Но для trap-сообщений используется 162 порт. Возможные команды протокола SNMPv3 приведены в таблице 9.1.

Таблица 1 - Основные команды протокола SNMPv3

Команда SNMP	Тип PDU	Назначение
GET-request	0	Получить значение указанной переменной или информацию о состоянии сетевого элемента.
GET-next-request	1	Получить значение переменной, не зная точного её имени (следующий логический идентификатор на дереве MIB).
SET-request	2	Присвоить переменной соответствующее значение. Используя для описания действия, которое должно быть выполнено.
GET-response	3	Отклик на GET-request, GET-next-request и SET-request. Содержит также информацию о состоянии (коды ошибок и другие данные).
TRAP	4	Отклик сетевого объекта на событие или на изменение состояния.
GetBulkRequest	5	Запрос пересылки больших объемов данных, например, таблиц.
InformRequest	6	Менеджер обращает внимание партнёра на определенную информацию в MIB.
SNMPv3-Trap	7	Отклик на событие (расширение по отношению к v1 и v2).
Report	8	Отчёт (функция пока не задана).

4. Формат SNMP-сообщения

Полное описание формата сообщения протокола SNMPv3 дано в документе RFC-3412 в разделе 6 «The SNMPv3 Message Format». Формат сообщения представлен на рисунке 4.

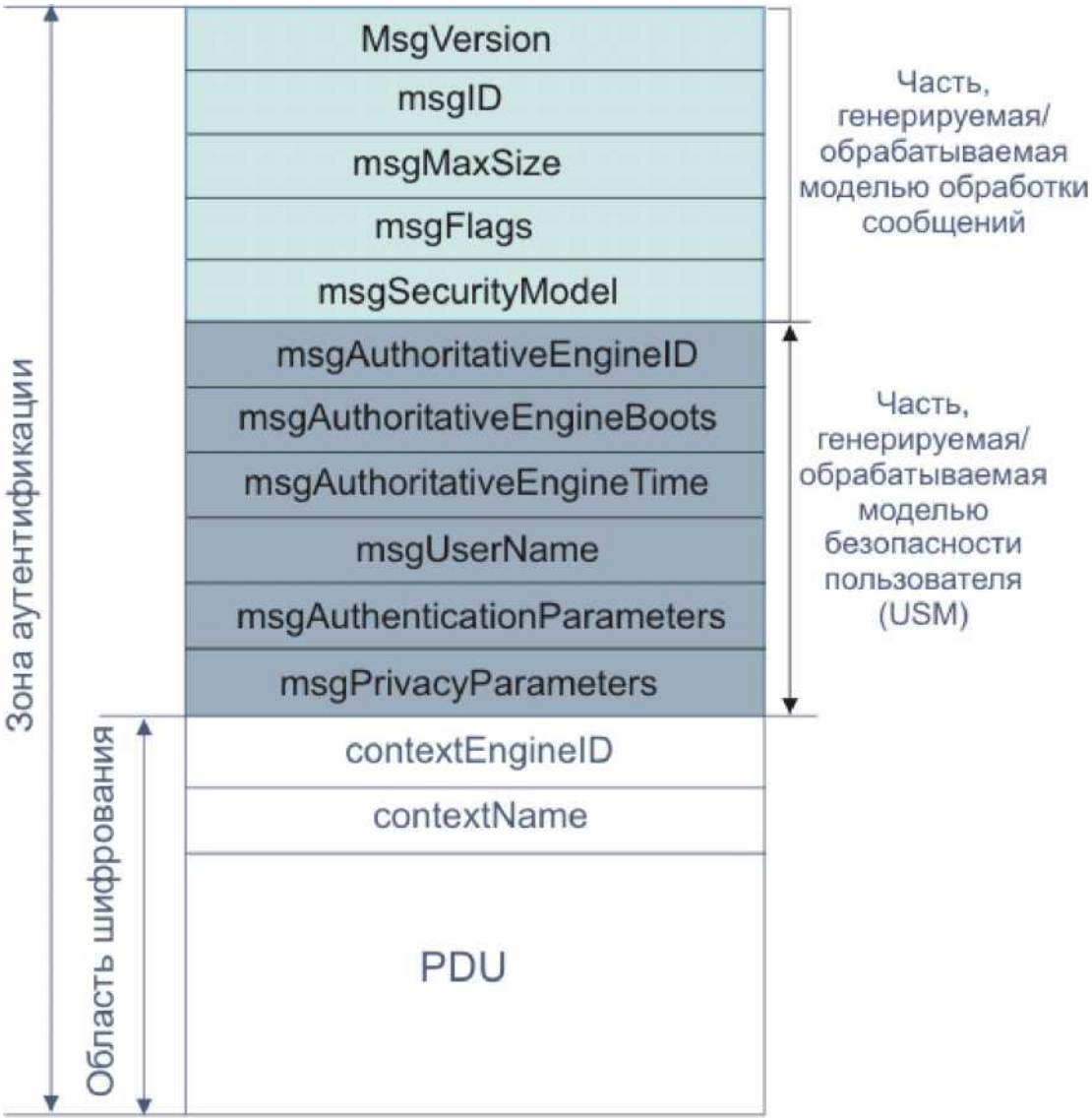


Рисунок 8.3 – Формат протокола SNMP

SNMP-сообщение логически разделено на три части:

Часть, которая формируются отправителем в рамках модели обработки сообщений и обрабатываются получателем.

Часть, отвечающая за функции безопасности.

Собственно поле данных.

В данном разделе дано описание полей первой и третей частей. Описание полей безопасности дано в разделе 6.5.

4. Протокол SSH.

SSH (Secure Shell) - это сетевой протокол, используемый для удаленного управления компьютером и для передачи файлов. SSH похож по функциональности на протоколы telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли. SSH позволяет передавать через безопасный канал любой другой сетевой протокол. Также, SSH может

использовать сжатие передаваемых данных для последующей их шифрации.

Первая версия протокола, SSH-1, была разработана в 1995 году исследователем Tatu Ylonen из Технологического университета Хельсинки, Финляндия. SSH-1 был написан для обеспечения большей конфиденциальности, чем протоколы rlogin, telnet и rsh. В 1996 году была разработана более безопасная версия протокола, SSH-2, уже несовместимая с SSH-1. Протокол приобрел еще большую популярность и к 2000 году его использовало уже порядка двух миллионов пользователей. В 2006 году протокол был утвержден рабочей группой IETF в качестве Интернет-стандарта.

В первой версии протокола есть существенные недостатки, поэтому в настоящее время SSH-1 практически нигде не применяется, наверное, вы помните знаменитый эпизод из фильма "Матрица-2: Перезагрузка", в котором Тринити использует эксплоит SSHNuke для взлома городской электростанции? Вначале она нашла уязвимый SSH-сервер, просканировав сеть программой Nmap. Затем, через "дыру" "SSH1 CRC2", она получила максимальные права доступа к этому серверу.

Многие взломщики сканируют сеть в поиске открытого порта SSH. Поэтому в целях безопасности необходимо запрещать доступ по ssh для суперпользователя. Обычно злоумышленники подбирают именно пароль суперпользователя. Протокол SSH-2 устойчив к атакам "man-in-middle", в отличие от протокола telnet. То есть, прослушивание трафика, "снифинг", ничего не дает злоумышленнику. Протокол SSH-2 также устойчив к атакам путем присоединения посредине (session hijacking) и обманом сервера имен (DNS spoffing).

Далее по тексту под SSH имеется ввиду именно вторая версия протокола, SSH-2.

Для работы по SSH нужен SSH-сервер и SSH-клиент. Сервер прослушивает соединения от клиентских машин и при установлении связи производит аутентификацию, после чего начинает обслуживание клиента. Клиент используется для входа на удаленную машину и выполнения команд.

SSH - это протокол прикладного уровня. SSH-сервер обычно слушает соединения на TCP-порту номер 22. Спецификация протокола SSH-2 содержится в RFC 4251. Для аутентификации сервера SSH использует алгоритм Diffie-Hellman'a. Для аутентификации клиента - шифрование с открытым ключом (оно сравнительно медленное). Для шифрования передаваемых данных - симметричное шифрование (оно более быстрое). Среди алгоритмов шифрования с открытым ключом чаще всего используются RSA и DSA. Из симметричных алгоритмов - AES, Blowfish и 3DES. Целостность переданных данных проверяется с помощью CRC32 в SSH1 или HMAC-SHA1/HMAC-MD5 в SSH2. Для сжатия шифруемых данных используется алгоритм LempelZiv (LZ77), обеспечивает такую же компрессию, что и архиватор zip.

Для подключения клиента требуется сгенерировать пару из открытого и

закрытого ключей. Если используется PuTTY под Windows, то это делается утилитой puttygen.exe. Под Linux обычно используется команда puttygen (для PuTTY) или ssh-keygen (для OpenSSH) и далее указывается, где находится закрытый ключ, а затем производится соединение с вводом пароля. Возможно также беспарольное соединение.

Наконец, несколько слов о кодировании сетевых адресов. Поскольку DNS - «ненадежный» протокол, в ssh он не используется. При адресации применяются IP-адреса, причем в проекте заложена поддержка IPv6. Все сообщения (пакеты) ssh содержат номер сообщения число от 1 до 255. Этот диапазон разбит на под интервалы, причем разным уровням протокола ssh отвечают разные диапазоны.

Запрещение удаленного root-доступа.

Запрещение подключения с пустым паролем или отключение входа по паролю.

Выбор нестандартного порта для SSH-сервера.

Ограничение списка IP-адресов, с которых разрешен доступ.

Запрещение доступа с некоторых, потенциально опасных адресов.

Отказ от использования распространенных или широко известных системных логинов для доступа по SSH.

Регулярный просмотр сообщений об ошибках аутентификации.

Установка систем обнаружения атак и вторжений.

Использование ловушек, подделывающих SSH-сервис.

Контрольные вопросы.

1. Что за протокол SSL/TLS и для чего используется;
2. В чем разница протокола ESP от протокола AH;
3. В чем разница протокола PPTP от протокола L2TP;
4. Что такое TKIP и как он создается;
5. Как осуществляется ассоциация обеспечения безопасности.

Основная:

1. НПП «Учтех-Профи» - Сетевая безопасность – Теория 2012г.
2. Олифер В.Г. Компьютерные сети –М., 2007г.
3. Принципы построения инфокоммуникационных сетей Санкт-Петербург -2013г.
4. Ургенишбаев К.М., Суйеубаев О.Б. Методические указания к выполнению лабораторных работ, АУЭС, Алматы, 2016г.
5. Межсетевой экран D-Link Руководство пользователь DFL-210/800/1600/2500, 2010г
6. Гордейчик С.В. Безопасность беспроводных сетей.-М.,2008
7. Мельников В.П. Информационная безопасность и защита информации.-М.,2008