Частное профессиональное образовательное учреждение
«Северо-Кавказский колледж инновационных технологий»

Курс лекций

Организация сетевого администрирования

Специальность: 29.02.06 Сетевое и системное администрирование

Наименование профиля: Технический

Курс лекций

«Организация сетевого администрирования»

Часть 1.

Лекция 1 Сетевые операционные системы

Тема: Сетевые операционные системы: структура, назначение, функции

1. Что такое сетевое программное обеспечение?
Сетевое программное обеспечение предназначено для организации совместной работы группы пользователей на разных компьютерах. Позволяет организовать общую файловую структуру, общие базы данных, доступные каждому члену группы. Обеспечивает возможность передачи сообщений и работы над общими проектами, возможность разделения ресурсов.
2. Сетевые операционные системы (Network Operating System – NOS) – это комплекс программ, обеспечивающих обработку, хранение и передачу данных в сети.
Сетевая операционная система выполняет функции прикладной платформы, предоставляет разнообразные виды сетевых служб и поддерживает работу прикладных процессов, выполняемых в абонентских системах. Сетевые операционные системы используют клиент-серверную, либо одноранговую архитектуру. Компоненты NOS располагаются на всех рабочих станциях, включенных в сеть.
NOS определяет взаимосвязанную группу протоколов верхних уровней, обеспечивающих выполнение основных функций сети. К ним, в первую очередь, относятся:
1. адресация объектов сети;
2. функционирование сетевых служб;
3. обеспечение безопасности данных;
4. управление сетью.

При выборе NOS необходимо рассматривать множество факторов. Среди них:
¬ набор сетевых служб, которые предоставляет сеть;
¬ возможность наращивания имен, определяющих хранимые данные и прикладные программы;
¬ механизм рассредоточения ресурсов по сети;

¬ способ модификации сети и сетевых служб;
¬ надежность функционирования и быстродействие сети;
¬ используемые или выбираемые физические средства соединения;
¬ типы компьютеров, объединяемых в сеть, их операционные системы;
¬ предлагаемые системы, обеспечивающие управление сетью;
¬ используемые средства защиты данных;
¬ совместимость с уже созданными прикладными процессами;
¬ число серверов, которое может работать в сети;
¬ перечень ретрансляционных систем, обеспечивающих сопряжение локальных сетей с различными территориальными сетями;
¬ способ документирования работы сети, организация подсказок и поддержек.

3. Функции и характеристики сетевых операционных систем (ОС).

Различают ОС со встроенными сетевыми функциями и оболочки над локальными ОС. По другому признаку классификации различают сетевые ОС одноранговые и функционально несимметричные (для систем "клиент/сервер").

Основные функции сетевой ОС:
1. управление каталогами и файлами;
2. управление ресурсами;
3. коммуникационные функции;
4. защита от несанкционированного доступа;
5. обеспечение отказоустойчивости;
6. управление сетью.

Управление каталогами и файлами в сетях заключается в обеспечении доступа к данным, физически расположенным в других узлах сети. Управление осуществляется с помощью специальной сетевой файловой системы. Файловая система позволяет обращаться к файлам путем применения привычных для локальной работы языковых средств. При обмене файлами должен быть обеспечен необходимый уровень конфиденциальности обмена (секретности данных).

Управление ресурсами включает обслуживание запросов на предоставление ресурсов, доступных по сети.

Коммуникационные функции обеспечивают адресацию, буферизацию, выбор направления для движения данных в разветвленной сети (маршрутизацию), управление потоками данных и др. Защита от несанкционированного доступа — важная функция, способствующая поддержанию целостности данных и их конфиденциальности. Средства защиты могут разрешать доступ к определенным данным только с некоторых терминалов, в оговоренное время, определенное число раз и т.п.

У каждого пользователя в корпоративной сети могут быть свои права доступа с ограничением совокупности доступных директорий или списка возможных действий, например, может быть запрещено изменение содержимого некоторых файлов.

Отказоустойчивость характеризуется сохранением работоспособности системы при воздействии дестабилизирующих факторов. Отказоустойчивость обеспечивается применением для серверов автономных источников питания, отображением или дублированием информации в дисковых накопителях. Под отображением обычно понимают наличие в системе двух копий данных с их расположением на разных дисках, но подключенных к одному контроллеру. Дублирование отличается тем, что для каждого из дисков с копиями используются разные контроллеры. Очевидно, что дублирование более надежно. Дальнейшее повышение отказоустойчивости связано с дублированием серверов, что однако требует дополнительных затрат на приобретение оборудования.

Управление сетью связано с применением соответствующих протоколов управления. Программное обеспечение управления сетью обычно состоит из менеджеров и агентов. Менеджером называется программа, вырабатывающая сетевые команды. Агенты представляют собой программы, расположенные в различных узлах сети. Они выполняют команды менеджеров, следят за состоянием узлов, собирают информацию о параметрах их функционирования, сигнализируют о происходящих событиях, фиксируют аномалии, следят за трафиком, осуществляют защиту от вирусов. Агенты с достаточной степенью интеллектуальности могут участвовать в восстановлении информации после сбоев, в корректировке параметров управления и т.п.

4. Структура сетевой операционной системы

Сетевая операционная система составляет основу любой вычислительной сети. Каждый компьютер в сети автономен, поэтому под сетевой операционной системой в широком смысле понимается совокупность операционных систем отдельных компьютеров, взаимодействующих с целью обмена сообщениями и разделения ресурсов по единым правилам – протоколам. В узком смысле сетевая ОС – это операционная система отдельного компьютера, обеспечивающая ему возможность работать в сети.

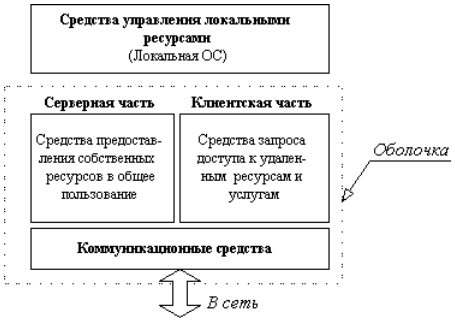

Рис.1 Структура сетевой ОС

В соответствии со структурой, приведенной на рис. 1, в сетевой операционной системе отдельной машины можно выделить несколько частей.

1. Средства управления локальными ресурсами компьютера: функции распределения оперативной памяти между процессами, планирования и диспетчеризации процессов, управления процессорами, управления периферийными устройствами и другие функции управления ресурсами локальных ОС.
2. Средства предоставления собственных ресурсов и услуг в общее пользование – серверная часть ОС (сервер). Эти средства обеспечивают, например, блокировку файлов и записей, ведение справочников имен сетевых ресурсов; обработку запросов удаленного доступа к собственной файловой системе и базе данных; управление очередями запросов удаленных пользователей к своим периферийным устройствам.
3. Средства запроса доступа к удаленным ресурсам и услугам – клиентская часть ОС (редиректор). Эта часть выполняет распознавание и перенаправление в сеть запросов к удаленным ресурсам от приложений и пользователей. Клиентская часть также осуществляет прием ответов от серверов и преобразование их в локальный формат, так что для приложения выполнение локальных и удаленных запросов неразличимо.
4. Коммуникационные средства ОС, с помощью которых происходит обмен сообщениями в сети. Эта часть обеспечивает адресацию и буферизацию сообщений, выбор маршрута передачи сообщения по сети, надежность передачи и т.п., т. е. является средством транспортировки сообщений.

5. Клиентское программное обеспечение

Для работы с сетью на клиентских рабочих станциях должно быть установлено клиентское программное обеспечение. Это программное обеспечение обеспечивает доступ к ресур-сам, расположенным на сетевом сервере. Тремя наиболее важными компонентами клиентского программного обеспечения являются редиректоры (redirector), распределители (desig-nator) и имена UNC (UNC pathnames).

Редиректоры

Редиректор – сетевое программное обеспечение, которое принимает запросы вво-да/вывода для удаленных файлов, именованных каналов или почтовых слотов и затем пере-назначает их сетевым сервисам другого компьютера. Редиректор перехватывает все запросы, поступающие от приложений, и анализирует их.

Фактически существуют два типа редиректоров, используемых в сети:

- клиентский редиректор (client redirector)
- серверный редиректор (server redirector).

Оба редиректора функционируют на представительском уровне модели OSI. Когда клиент делает запрос к сетевому приложению или службе, редиректор перехватывает этот запрос и проверяет, является ли ресурс локальным (находящимся на запрашивающем ком-пьютере) или удаленным (в сети).Если редиректор определяет, что это локальный запрос, он направляет запрос центральному процессору для немедленной обработки. Если запрос пред-назначен для сети, редиректор направляет запрос по сети к соответствующему серверу. По существу, редиректоры скрывают от пользователя сложность доступа к сети. После того как сетевой ресурс определен, пользователи могут получить к нему доступ без знания его точно-го расположения.

Распределители

Распределитель (designator) представляет собой часть программного обеспечения, управляющую присвоением букв накопителя (drive letter) как локальным, так и удаленным сетевым ресурсам или разделяемым дисководам, что помогает во взаимодействии с сетевыми ресурсами. Когда между сетевым ресурсом и буквой локального накопителя создана ассоциация, известная также как отображение дисковода (mapping a drive), распределитель отслеживает присвоение такой буквы дисковода сетевому ресурсу. Затем, когда пользователь или приложение получат доступ к диску, распределитель заменит букву дисковода на сете-вой адрес ресурса, прежде чем запрос будет послан редиректору.

Имена UNC

Редиректор и распределитель являются не единственными методами, используемыми для доступа к сетевым ресурсам. Большинство современных сетевых операционных систем, так же как и Windows 95, 98, NT, распознают имена UNC (Universal Naming Convention — Универсальное соглашение по наименованию). UNC представляют собой стандартный спо-соб именования сетевых ресурсов. Эти имена имеют форму \\Имя_сервера\имя_ресурса. Способные работать с UNC приложения и утилиты командной строки используют имена UNC вместо отображения сетевых дисков.

6. Серверное программное обеспечение

Для того чтобы компьютер мог выступать в роли сетевого сервера необходимо установить серверную часть сетевой операционной системы, которая позволяет поддерживать ресурсы и распространять их среди сетевых клиентов. Важным вопросом для сетевых серверов является возможность ограничить доступ к сетевым ресурсам. Это называется сетевой защитой (network security). Она предоставляет средства управления над тем, к каким ресурсам могут получить доступ пользователи, степень этого доступа, а также, сколько пользователей смогут получить такой доступ одновременно.

Этот контроль обеспечивает конфиденциальность и защиту и поддерживает эффективную сетевую среду.
В дополнение к обеспечению контроля над сетевыми ресурсами сервер выполняет следующие функции:
¬ предоставляет проверку регистрационных имен (logon identification) для пользователей;
¬ управляет пользователями и группами;
¬ хранит инструменты сетевого администрирования для управления, контроля и аудита;
¬ обеспечивает отказоустойчивость для защиты целостности сети.

7. Клиентское и серверное программное обеспечение

Некоторые из сетевых операционных систем, в том числе Windows NT, имеют программные компоненты, обеспечивающие компьютеру какклиентские, так и серверные возможности. Это позволяет компьютерам поддерживать и использовать сетевые ресурсы и преобладает в одноранговых сетях. В общем, этот тип сетевых операционных систем не так мощен и надежен, как законченные сетевые операционные системы. Главное преимущество комбинированной клиентско–серверной сетевой операционной системы заключается в том, что важные ресурсы, расположенные на отдельной рабочей станции, могут быть разделены с остальной частью сети. Недостаток состоит в том, что если рабочая станция поддерживает много активно используемых ресурсов, она испытывает серьезное падение производительности. Если такое происходит, то необходимо перенести эти ресурсы на сервер для увеличения общей производительности.
В зависимости от функций, возлагаемых на конкретный компьютер, в его операционной системе может отсутствовать либо клиентская, либо серверная части.
На рис. 2 компьютер 1 выполняет функции клиента, а компьютер 2 – функции сервера, соответственно на первой машине отсутствует серверная часть, а на второй - клиентская.

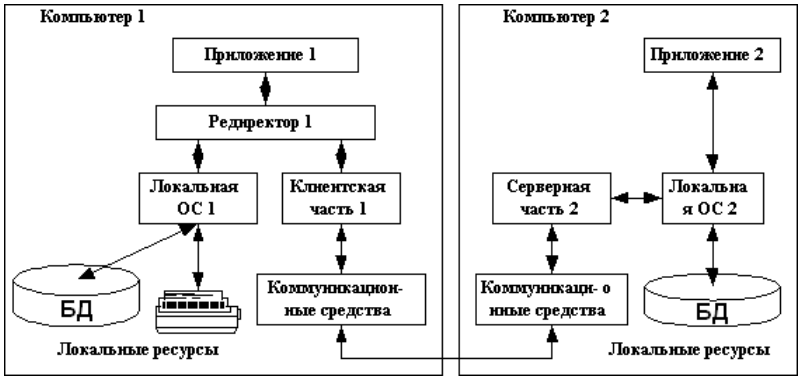

Рис. 2 Взаимодействие компонентов NOS

Если выдан запрос к ресурсу данного компьютера, то он переадресовывается локальной операционной системе. Если же это запрос к удаленному ресурсу, то он переправляется в клиентскую часть, где преобразуется из локальной формы в сетевой формат, и передается коммуникационным средствам. Серверная часть ОС компьютера 2 принимает запрос, преобразует его в локальную форму и передает для выполнения своей локальной ОС. После того, как результат получен, сервер обращается к транспортной подсистеме и направляет ответ клиенту, выдавшему запрос. Клиентская часть преобразует результат в соответствующий формат и адресует его тому приложению, которое выдало запрос.

8. Требования к современным операционным системам

Главным требованием, предъявляемым к операционной системе, является выполнение ею основных функций эффективного управления ресурсами и обеспечение удобного интерфейса для пользователя и прикладныхпрограмм. Современная ОС, как правило, должна поддерживать мультипрограммную обработку, виртуальную память, свопинг, многооконный графический интерфейс пользователя, а также выполнять многие другие необходимые функции и услуги. Кроме этих требований функциональной полноты к операционным системам предъявляются не менее важные эксплуатационные требования, которые перечислены ниже.

Расширяемость. В то время как аппаратная часть компьютера устаревает за несколько лет, полезная жизнь операционных систем может измеряться десятилетиями. Примером может служить ОС UNIX. Поэтому операционные системы всегда изменяются со временем эволюционно, и эти изменения более значимы, чем изменения аппаратных средств. Изменения ОС обычно заключаются в приобретении ею новых свойств, например поддержке новых типов внешних устройств или новых сетевых технологий. Если код ОС написан таким образом, что дополнения и изменения могут вноситься без на¬ рушения целостности системы, то такую ОС называют расширяемой. Расширяемость достигается за счет модульной структуры ОС, при которой про¬граммы строятся из набора отдельных модулей, взаимодействующих только через функциональный интерфейс.

Переносимость. В идеале код ОС должен легко переноситься с процессора одного типа на процессор другого типа и с аппаратной платформы (которые различаются не только типом процессора, но и способом организации всей аппаратуры компьютера) одного типа на аппаратную платформу другоготипа. Переносимые ОС имеют несколько вариантов реализации для разных платформ, такое свойство ОС называют также многоплатформенностью.

Совместимость. Существует несколько «долгоживущих» популярных операционных систем (разновидности UNIX, MS-DOS, Windows 3.x, Windows NT, OS/2), для которых наработана широкая номенклатура приложений. Некоторые из них пользуются широкой популярностью. Поэтому для пользователя, переходящего по тем или иным причинам с

одной ОС на другую, очень привлекательна возможность запуска в новой операционной системе привычно¬го приложения. Если ОС имеет средства для выполнения прикладных про¬грамм, написанных для других операционных систем, то про нее говорят, что она обладает совместимостью с этими ОС. Следует различать совместимость на уровне двоичных кодов и совместимость на уровне исходных текстов. Понятие совместимости включает также поддержку пользовательских интерфейсов других ОС.

Надежность/ отказоустойчивость. Система должна быть защищена как от внутренних, так и от внешних ошибок, сбоев и отказов. Ее действия должны быть всегда предсказуемыми, а приложения не должны иметь возможности наносить вред ОС. Надежность и отказоустойчивость ОС прежде всего определяются архитектурными решениями, положенными в ее основу, а также качеством ее реализации (отлаженностью кода). Кроме того, важно, включает ли ОС программную поддержку аппаратных средств обеспечения отказоустойчивости, таких, например, как дисковые массивы или источники бесперебойного питания.

Безопасность. Современная ОС должна защищать данные и другие ресурсы вычисли-тельной системы от несанкционированного доступа. Чтобы ОС обладала свойством безопасности, она должна как минимум иметь в своем составе средства аутентификации — определения легальностипользователей, авторизации — предоставления легальным пользователям дифференцированных прав доступа к ресурсам, аудита — фиксации всех «подозрительных» для безопасности системы событий. Свойство безопасности особенно важно для сетевых ОС. В таких ОС к задаче контроля доступа добавляется задача защиты данных, передаваемых по сети.

Производительность. Операционная система должна обладать настолько хорошим быстродействием и временем реакции, насколько это позволяет аппаратная платформа. На производительность ОС влияет много факторов, среди которых основными являются архитектура ОС, многообразие функций, качество программирования кода, возможность исполнения ОС на высокопроизводительной (многопроцессорной) платформе.

9. Выбор сетевой операционной системы

При выборе сетевой операционной системы необходимо учитывать:

¬ совместимость оборудования;
¬ тип сетевого носителя;
¬ размер сети;
¬ сетевую топологию;
¬ требования к серверу;
¬ операционные системы на клиентах и серверах;
¬ сетевая файловая система;
¬ соглашения об именах в сети;
¬ организация сетевых устройств хранения.

В настоящее время наибольшее распространение получили три основные сетевые ОС — UNIX, Windows NT и Novell Netware.

ОС UNIX применяют преимущественно в крупных корпоративных сетях, поскольку эта система характеризуется высокой надежностью, возможностью легкого масштабирования сети. В Unix имеется ряд команд и поддерживающих их программ для работы в сети. Во-первых, это команды ftp, telnet, реализующие файловый обмен и эмуляцию удаленного узла на базе протоколов TCP/IP. Во-вторых, протокол, команды и программы UUCP, разработанные с ориентацией на асинхронную модемную связь по телефонным линиям между удаленными Unix-узлами в корпоративных и территориальных сетях.

ОС Windows NT включает серверную (Windows NT Server) и клиентскую (Windows NT Workstation) части и, тем самым, обеспечивает работу в сетях "клиент/сервер". Windows NT обычно применяют в средних по масштабам сетях.

ОС Novell Netware состоит из серверной части и оболочек Shell, размещаемых в кли-ентских узлах. Предоставляет пользователям возможность совместно использовать файлы, принтеры и другое оборудование. Содержит службу каталогов, общую распределённую базу данных пользователей и ресурсов сети. Эту ОС чаще применяют внебольших сетях.

Лекция 2 Структура сетевой операционной системы

Тема: Структура и основные компоненты сетевых ОС. Функции по управлению сетевыми и локальными ресурсами.

Управление использованием ресурсов – одна из основных задач ОС.

ОС должна управлять использованием ресурсов вычислительной машины таким образом, чтобы обеспечить максимальную эффективность ее функционирования. Критерием эффективности может быть, например, пропускная способность или реактивность (под реактивностью понимают скорость реакции) системы. Управление ресурсами включает решение двух общих, не зависящих от типа ресурса задач:

- планирование использования ресурса, а именно - определение какому процессу, когда, в каком объеме, необходимо выделить данный ресурс;
- отслеживание состояния ресурса, то есть поддерживать набор оперативной информации о степени занятости ресурса.

Сетевая ОС (СОС) позволяет разделять ресурсы не только локально, но и в рамках сети объединяющей машины со своими средствами межсетевого взаимодействия. Она обязательно содержит программную поддержку для сетевых интерфейсных устройств, а также средства для удаленного входа в другие компьютеры сети и средства доступа к удаленным ресурсам, однако эти дополнения существенно не меняют структуру самой операционной системы. Фактически на современном уровне развития компьютерных технологий наличие у ОС возможностей для сетевого взаимодействия из разряда желательного перешло в разряд необходимого для полноценной работы пользователя. В отличии от СОС, распределенные ОС (РОС), реализует сетевое разделения ресурсов, моделируя единую виртуальную машину над сетью. Работая с РОС пользователю нет необходимости знать подключена его машина к сети, является ли данный ресурс локальным и где на планете выполняется его программа. Основное отличие СОС от РОС – в сети взаимодействуют несколько СОС (по одной на абонента), в то время как в РОС – есть одна операционная система, которая скрывает сеть.

Набор характеристик СОС.

Многопользовательские: позволяют 2 и более пользователям запускать программы в рамках одной ОС. Таким образом ОС UNIX многопользовательские, а Windows NT - нет. Последняя не позволяет нескольким пользователям одновременно работать (запускать свои приложения). В NT предоставление возможности использования мощностей процессора нескольким пользователям одновременно перекладывается с ОС на программистов (например используется технология клиент-сервер)

1) Поддерживающие многопроцессорность: Последняя может быть симметричной (процессоры равномерно нагружаются кодами разных программ), асимметричной (один процессор выполняет один процесс).

2) Многозадачность: Многозадачная ОС управляет ресурсами разделяемыми несколькими одновременно выполняющимися конкурирующими программами. Многозадачность разделяется на несколько типов, в зависимости от реализованного алгоритма управления разделением процессорного времени. Основные виды многозадачности – вытесняющая (ОС выделяет квант времени процессу или нити, затем прерывает их выполнение и выделяет квант времени следующему процессу или нити) и кооперативная (сам процесс определяет в какой момент времени вернуть ОС управление, например во время ожидания ввода)
3) Многонитеевые: Позволяет распараллеливать вычисления в рамках одного процесса. С точки зрения программирования нить – информация о состоянии (контексте) процесса. Нить создается и используется таким образом, что несколько процессов (нитей) может выполняться в рамках одного кода, но с использованием разных данных об окружении (контекстах). Обычно многонитиевость используется при написании программ-серверов, которым надо взаимодействовать единым образом с заранее неизвестным числом пользователей.
ОС делятся по критерию оптимизации на системы:
1) Пакетные: критерий эффективности – максимальное число решенных задач, которые поступают в ОС наборами (пакетами). ОС оптимизирует выполнение задач, а не взаимодействие с пользователем.
2) Реального времени: Отвечают на входящие данные немедленно. Критерий эффективности – гарантированное время реакции системы (скорость выполнения процессов и разделение процессорного времени) на информационный сигнал. Неспециализированные UNIX и DOS ОС не являются системами реального времени, т.к. не гарантируют одинаковое время реакции системы на входные данные.
3) Разделения времени: процессорное время равномерно распределяется между задачами, что делает работу пользователя более удобной. Критерий оптимальности – честное распределение (по потребностям) процессорного времени между разными задачами с одинаковым приоритетом на использование этого ресурса.
Большинство СОС можно отнести к последним двум типам.
Также сетевые ОС делятся на СОС со встроенными сетевыми функциями и на оболочки с сетевыми функциями над локальными ОС.

Набор критериев.
Рассмотрим набор критериев, на основе которого решается на сколько хорошо данная ОС может выполнять функции СОС.
Основные требования предъявляемые фирмами к современным СОС:
1) Системная архитектура – управление какими ресурсами и какие алгоритмы управления она поддерживает, можно ли ее запустить на многопроцессорной архитектуре, какие микропроцессорные архитектуры (Intel x86, Alpha, PowerPC) она поддерживает

2) Масштабируемость – количество ресурсов, которыми сможет управлять ОС (вдруг ваша распределенная гигабайтовая БД станет терабайтовой или количество одновременно открытых TCP соединений увеличиться на порядок)
3) Производительность – скорость выполнения СОС требуемого класса задач, количество одновременных обращений пользовательских процессов которое в состоянии обслужить система
4) Надежность – поддержка средствами СОС средств резервирования данных, транзакций, поддержка или нахождение в составе СОС надежной файловой системы.
5) Безопасность – какой уровень защиты информации поддерживает СОС, ограничения на доступ к каким ресурсам она поддерживает, какая система прав доступа поддерживается.
6) Средства администрирования – какие утилиты и как помогают администировать СОС
7) Поддержка сетевых сред – поддерживает ли СОС физические устройства работающие с Ethernet, Token ring, оптоволокном и т.п.
8) Поддержка стеков протоколов – на каких и скольких стеках протоколов может функционировать СОС и поддержка программного обеспечения для работы с данными в рамках всемирной сети Интернет
9) Сетевая печать – сколько поддерживается средствами СОС принтеров на сервер, очередей на принтер
10) Приложения – какие приложения включены в стандартную поставку СОС, какую минимальную функциональность обеспечивает СОС (это могут быть почтовые сервера и клиенты, средства разработки, сервера печати, Интернет сервера и т.п.)
11) Совместимость – на сколько СОС совместима с уже имеющимися программно-аппаратными комплексами предприятия.
Исходя из описанных выше требований можно заключить, что хорошо спроектированная СОС должна:
• поддерживать возможность работы на многопроцессорной ЭВМ (с симметричной многопроцессорностью)
• быть многозадачной и поддерживать нити в рамках одного процесса.
• при необходимости быть многопользовательской.
В общем случае для большинства современных коммерческих СОС вопрос, какая из них лучше задавать бессмысленно – раз все выдерживают конкуренцию, значит у каждой есть какие-то достоинства. Решение о выборе СОС обычно основываются на оценке набора критериев, подобно приведенному выше, применительно к конкретной ситуации.
Структура СОС
Каждый компьютер с установленной СОС в значительной степени автономен, поэтому под сетевой операционной системой в широком смысле понимается совокупность операционных систем отдельных компьютеров, взаимодействующих с целью обмена сообщениями и разделения ресурсов по единым правилам - протоколам. В узком смысле сетевая ОС – это

операционная система отдельного компьютера, обеспечивающая ему возможность работать в сети.

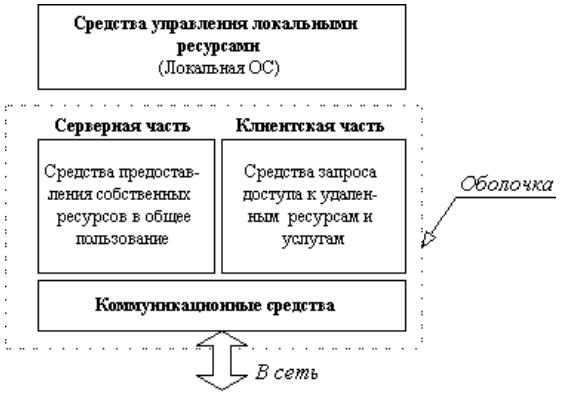

Рис.1

В сетевой операционной системе отдельной машины можно выделить несколько частей (рисунок 1):

1) Средства управления локальными ресурсами компьютера: функции распределения оперативной памяти между процессами, функции планирования и диспетчеризации процессов, управления процессорами в мультипроцессорных машинах, управления периферийными устройствами и другие функции управления ресурсами локальных ОС.

2) Средства предоставления собственных ресурсов и услуг в общее пользование - серверная часть ОС (сервер). Эти средства обеспечивают, например, блокировку файлов и записей, что необходимо для их совместного использования; ведение справочников имен сетевых ресурсов; обработку запросов удаленного доступа к собственной файловой системе и базе данных; управление очередями запросов удаленных пользователей к своим периферийным устройствам.

3) Средства запроса доступа к удаленным ресурсам и услугам и их использования - клиентская часть ОС. Эта часть выполняет распознавание и перенаправление в сеть запросов к удаленным ресурсам от приложений и пользователей, при этом запрос поступает от приложения в локальной форме, а передается в сеть в форме, соответствующей требованиям сервера. Клиентская часть также осуществляет прием ответов от серверов и преобразование их в локальный формат, так что для приложения выполнение локальных и удаленных запросов неразличимо.

4) Коммуникационные средства ОС, с помощью которых происходит обмен сообщениями между СОС в сети. Эта часть обеспечивает адресацию и буферизацию сообщений, выбор маршрута передачи сообщения по сети, надежность передачи и т.п., то есть является средством транспортировки сообщений.

Все множество СОС можно разбить на две группы:

1) Первые сетевые ОС представляли собой совокупность неспециализированной (General) ОС и надстройки, добавляющей к ней сетевые функции (рисунок 2).

2) Однако более эффективным представляется путь разработки операционных систем, изначально предназначенных для работы в сети. Сетевые функции у ОС такого типа встроены в основные модули системы, что обеспечивает их логическую стройность, простоту эксплуатации и модификации, а также высокую производительность.

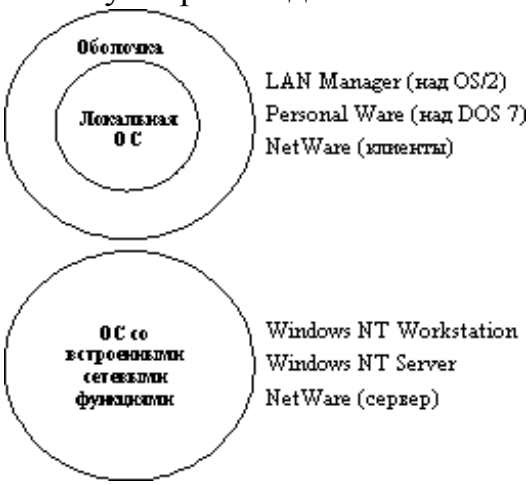

Рис.2

Как видно из структуры, СОС это - ОС с добавлением сетевых функций.

Основополагающим критерием по значительности влияния на производительность и масштабируемость операционной системы является ее архитектура. Операционные системы прошли длительный путь развития от монолитных систем к хорошо структурированным модульным системам, способным к развитию, расширению и обладающие хорошей переносимостью.

1. Монолитные системы

В общем случае "структура" монолитной системы представляет собой отсутствие структуры (рисунок 3). ОС написана как набор процедур, каждая из которых может вызывать другие, когда ей это нужно. При использовании этой техники каждая процедура системы имеет хорошо определенный интерфейс, и каждая может вызвать любую другую при необходимости.

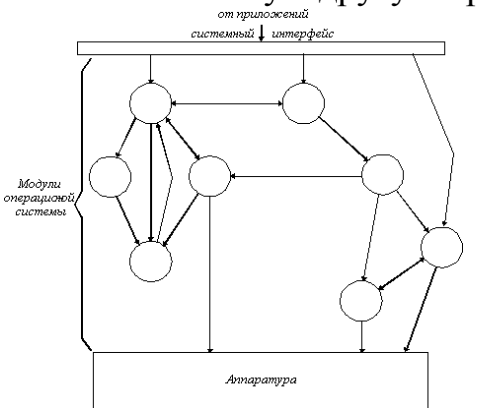

Рис. 3

Монолитная ОС собирается из программных модулей и затем компилируется как единая система. И хотя как программа такая СОС может

и быть модульной, на практике взаимодействие ее процедур происходит в единой области видимости и любая процедура может вызвать любую.

2. Многоуровневые системы

При структуризации от монолитных систем переходят к многоуровневым. Уровни образуются группами функций операционной системы - файловая система, управление процессами и устройствами и т.п. Каждый уровень может взаимодействовать только со своим непосредственным соседом - выше- или нижележащим уровнем. Прикладные программы или модули самой операционной системы передают запросы вверх и вниз по этим уровням (рисунок 4).

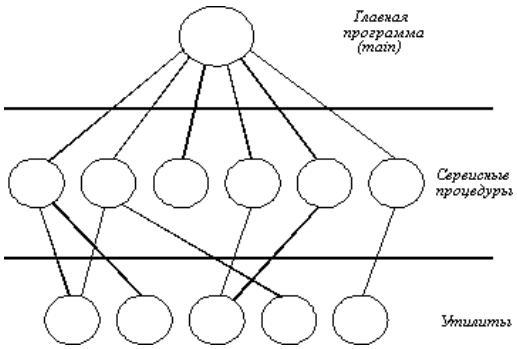

Рис. 4

Хотя такой структурный подход на практике обычно работал неплохо, сегодня он все больше воспринимается монолитным, старые ОС UNIX с многоуровневой структурой сейчас характеризуются как ОС с монолитными ядрами. В системах, имеющих многоуровневую структуру было нелегко удалить один слой и заменить его другим в силу множественности и размытости интерфейсов между слоями. Добавление новых функций и изменение существующих требовало хорошего знания операционной системы и массы времени. Когда стало ясно, что операционные системы живут долго и должны иметь возможности развития и расширения, монолитный подход сменился моделью клиент-сервер с тесно связанной с ней концепция микроядра.

3. Модель клиент-сервер и микроядра

Применительно к структурированию ОС идея использования взаимодействия клиент-сервер и микроядра состоит в разбиении ее на несколько процессов - серверов, каждый из которых выполняет отдельный набор сервисных функций - например, управление памятью, управление файловой системой. Каждый сервер выполняется в пользовательском режиме. Клиент, которым может быть либо другой компонент ОС, либо прикладная программа, запрашивает сервис, посылая сообщение на сервер. Ядро ОС (называемое здесь микроядром), работая в привилегированном режиме, доставляет сообщение нужному серверу, сервер выполняет операцию, после чего ядро возвращает результаты клиенту с помощью другого сообщения (рисунок 5).

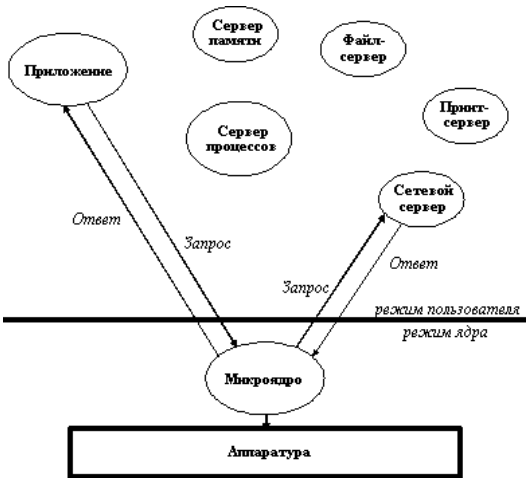

Рис. 5

Подход с использованием микроядра заменил вертикальное распределение функций операционной системы на горизонтальное. Компоненты, лежащие выше микроядра, хотя и используют сообщения, пересылаемые через микроядро, взаимодействуют друг с другом непосредственно. Это свойство микроядерных систем позволяет естественно использовать их в распределенных средах. При получении сообщения микроядро может его обработать или переслать другому процессу. Поскольку для микроядра безразлично, поступило ли сообщение от локального или удаленного процесса, подобная схема передачи сообщений является удобной основой удаленных вызовов процедур (RPC - remote procedure calls). Микроядро занимается основной функцией ОС – управлением ресурсами, зачастую оно берет на себя функции взаимодействия с аппаратурой, хотя предпочтительно в рамках микроядра выделять машиннозависимый функции в отдельные подмодули для улучшения переносимости. Различные варианты реализации модели клиент-сервер в структуре ОС могут существенно различаться по объему работ, выполняемых в режиме ядра. На одном краю этого спектра находится разрабатываемая фирмой IBM на основе микроядра Mach операционная система Workplace OS, придерживающаяся чистой микроядерной доктрины, состоящей в том, что все несущественные функции ОС должны выполняться не в режиме ядра, а в непривилегированном (пользовательском) режиме. На другом - Windows NT, в составе которой имеется исполняющая система (NT executive), работающая в режиме ядра и выполняющая функции обеспечения безопасности, ввода-вывода и другие.

Микроядерный подход при проектировании архитектуры ОС требует ответа на вопрос, какие функции ОС следует оставить в ядре, а какие вынести из него. Модули, содержащиеся в ядре, нельзя заменить без его перекомпиляции. Причем если само микроядро является плохоструктурированным, то замена одного его модуля на другой (например замена планировщика задач) может стать очень трудной задачей. С другой стороны хотя вынос за пределы ядра не основных и динамически изменяемых функций хотя и делает ОС хорошо масштабируемой и более надежной (ядро обычно выступает как единый домен сбоев, в то время как

гибель внешнего сервера ОС может пережить безболезненно), это сказывается на ее производительности.
В состав микроядра обычно не включают сетевые функции, пользовательский интерфейс, файловую систему, а лишь основные функции управления ???

Достоинства и недостатки микроядерного подхода
В настоящее время именно операционные системы, построенные с использованием модели клиент-сервер и концепции микроядра, в наибольшей степени удовлетворяют требованиям, предъявляемым ксовременным СОС.
• Высокая степень переносимости и совместимости обусловлена тем, что весь машинно-зависимый код изолирован в микроядре, поэтому для переноса системы на новый процессор требуется меньше изменений и все они логически сгруппированы вместе.
• Микроядерный подход позволяет легко перестраивать специализацию ОС. Является ли операционная система маленькой, как DOS, или большой, как UNIX, для нее неизбежно настанет необходимость приобрести свойства, не заложенные в ее конструкцию. Увеличивающаяся сложность монолитных операционных систем делала трудным, если вообще возможным, внесение изменений в ОС с гарантией надежности ее последующей работы. Ограниченный набор четко определенных интерфейсов микроядра открывает путь к упорядоченному росту и эволюции ОС. Обычно операционная система выполняется только в режиме ядра, а прикладные программы - только в режиме пользователя, за исключением тех случаев, когда они обращаются к ядру за выполнением системных функций. В отличие от обычных систем, система построенная на микроядре, выполняет свои серверные подсистемы в режиме пользователя, как обычные прикладные программы. Такая структура позволяет изменять и добавлять серверы, не влияя на целостность микроядра.
• Надежность микроядерной архитектуры выше, чем у монолитной. Микроядро легче тестировать, при этом оно выполняется в привилегированном, защищенном режиме процессоров и сбой внешних служб, выполняющихся в отдельных виртуальных машинах в непривилегированном режиме, не приведет к краху системы в целом. Одной из проблем традиционно организованных операционных систем является наличие множества интерфейсов прикладного программирования (API - Application Programming Interface), не все из которых хорошо документированы. В результате невозможно гарантировать правильность программ, использующих несколько API, и даже правильность работы самой операционной системы. Микроядро, обладающее небольшим набором API, увеличивает шансы получения качественных программ. Конечно, этот компактный интерфейс облегчает жизнь только системных программистов; прикладной программист по прежнему должен бороться с сотнями вызовов.

• Поддержка распределенных и сетевых приложений хорошо вписывается в концепцию микроядра, основанном на горизонтальному разделению служб ОС.

Основным недостатком использования микроядерного подхода на практике является снижение быстродействия на локальных задачах - замедление скорости выполнения системных вызовов при передаче сообщений через микроядро по сравнению с классическим подходом.

Основные ресурсы и службы СОС. Способы управления ими.

Важнейшей функцией операционной системы является организация рационального использования всех аппаратных и программных ресурсов системы. К основным ресурсам могут быть отнесены: процессоры, память (виртуальная память), внешние устройства.

Процесс - абстракция, описывающая выполняющуюся программу. Для операционной системы процесс представляет собой единицу выполнения и динамически изменяющуюся заявку на потребление системных ресурсов. Подсистема управления процессами планирует выполнение процессов, то есть распределяет процессорное время между несколькими одновременно существующими в системе процессами, а также занимается созданием и уничтожением процессов, обеспечивает процессы необходимыми системными ресурсами, поддерживает взаимодействие между процессами.

СОС реализует в этой подсистеме удаленное межпроцессное взаимодействие, работу процессов с удаленными ресурсами.

1. Планирование процессов

Планирование процессов включает в себя решение следующих задач:

1) определение момента времени для смены выполняемого процесса
2) выбор процесса на выполнение из очереди готовых процессов
3) переключение контекстов "старого" и "нового" процессов

Существует множество различных алгоритмов планирования процессов, по разному решающих вышеперечисленные задачи. Они преследуют различные цели и обеспечивают различное качество мультипрограммирования. Среди этого множества алгоритмов выделяются две группы наиболее часто встречающихся алгоритмов: алгоритмы, основанные на квантовании, и алгоритмы, основанные на приоритетах.

В соответствии с алгоритмами, основанными на квантовании, смена активного процесса происходит, если:

1) процесс завершился и покинул систему
2) произошла ошибка
3) процесс перешел в состояние ожидания
4) исчерпан квант процессорного времени, отведенный данному процессу

Процесс, который исчерпал свой квант, переводится в состояние готовность и ожидает, когда ему будет предоставлен новый квант процессорного времени, а на выполнение в соответствии с определенным правилом выбирается новый процесс из очереди готовых. Таким образом, ни один

процесс не занимает процессор надолго, поэтому квантование широко используется в системах разделения времени.
Приоритет может выражаться целыми или дробными, положительным или отрицательным значением.Чем выше привилегии процесса, тем меньше времени он будет проводить в очередях. Приоритет может назначаться директивно администратором системы в зависимости от важности работы или внесенной платы, либо вычисляться самой ОС по определенным правилам, он может оставаться фиксированным на протяжении всей жизни процесса либо изменяться во времени в соответствии с некоторым законом.В последнем случае приоритеты называются динамическими.
Существует две разновидности приоритетных алгоритмов: алгоритмы, использующие относительные приоритеты, и алгоритмы, использующие абсолютные приоритеты.
В обоих случаях выбор процесса на выполнение из очереди готовых осуществляется одинаково: выбирается процесс, имеющий наивысший приоритет. По разному решается проблема определения момента смены активного процесса. В системах с относительными приоритетами активный процесс выполняется до тех пор, пока он сам не покинет процессор, перейдя в состояние ожидания (или же произойдет ошибка, или процесс завершится). В системах с абсолютными приоритетами выполнение активного процесса прерывается еще при одном условии: если в очереди готовых процессов появился процесс, приоритет которого выше приоритета активного процесса. Во многих операционных системах алгоритмы планирования построены с использованием как квантования, так и приоритетов. Например, в основе планирования лежит квантование, но величина кванта и/или порядок выбора процесса из очереди готовых определяется приоритетами процессов.
Существует два основных типа процедур планирования процессов - вытесняющие (preemptive) и невытесняющие (non-preemptive).
Non-preemptive multitasking - невытесняющая многозадачность - это способ планирования процессов, при котором активный процесс выполняется до тех пор, пока он сам, по собственной инициативе, не отдаст управление планировщику операционной системы для того, чтобы тот выбрал из очереди другой, готовый к выполнению процесс. Программист должен обеспечить "дружественное" отношение своей программы к другим выполняемым одновременно с ней программам, достаточно часто отдавая имуправление. Крайним проявлением "недружественности" приложения является его зависание, которое приводит к общему краху системы. В системах с вытесняющей многозадачностью такие ситуации, как правило, исключены, так как центральный планирующий механизм снимет зависшую задачу с выполнения.
Preemptive multitasking - вытесняющая многозадачность - это такой способ, при котором решение о переключении процессора с выполнения одного процесса на выполнение другого процесса принимается планировщиком операционной системы, а не самой активной задачей.

Для сетевых ОС наиболее рациональным является вытесняющая многозадачность, которая гарантирует обработку сетевого взаимодействиясо временем реакции приближенным к системам реального времени.
Совместное использование ресурсов несколькими одновременно работающими процессами в рамках локальной ОС создает проблемы как синхронизации, так и взаимной блокировки ресурсов (для чего ОС должна реализовывать алгоритмы регламентирующие выделение ресурсов.

2. Управление памятью

Память, к которой может иметь доступ СОС может быть локальной, разделяемой, распределенной, для работы со всеми видами памяти в ОС создается менеджер памяти.
Функциями ОС по управлению памятью являются: отслеживание свободной и занятой памяти, выделение памяти процессам и освобождение памяти при завершении процессов, вытеснение процессов из оперативной памяти на диск, когда размеры основной памяти не достаточны для размещения в ней всех процессов, и возвращение их в оперативную память, когда в ней освобождается место, а также настройка адресов программы на конкретную область физической памяти.
Современная СОС должна уметь работать с виртуальной памятью, так как это позволяет оптимально использовать ресурс и добиваться увеличения быстродействия по сравнению с работой с физической памятью.
Виртуальная память - это совокупность программно-аппаратных средств, позволяющих пользователям писать программы, размер кода и данных которых превосходит имеющуюся оперативную память; для этого виртуальная память решает следующие задачи:
• размещает данные в запоминающих устройствах разного типа, например, часть программы в оперативной памяти, а часть на диске;
• перемещает по мере необходимости данные между запоминающими устройствами разного типа, например, подгружает нужную часть программы с диска в оперативную память;
• преобразует виртуальные адреса в физические.
Не вдаваясь в подробности можно заметить, что наиболее эффективные алгоритмы работы с памятью наиболее сложны в реализации. Наиболее оптимальны сегментно-страничная организация виртуальной памяти с использованием упреждающих алгоритмов подкачки и выталкивания страниц.
Также необходимо стремиться к оптимальному использованию кэширования данных (рис. 6 – иерархия ЗУ)

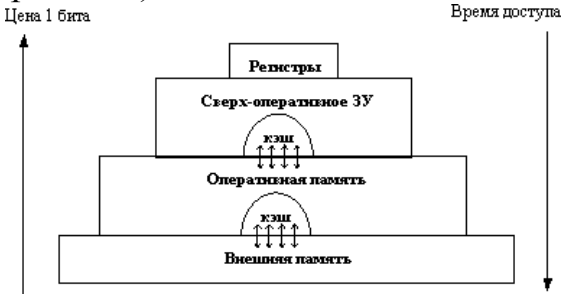

Рис. 6. (Иерархия ЗУ)

3. Управление вводом-выводом

Одной из главных функций ОС является управление всеми устройствами ввода-вывода компьютера. ОС должна передавать устройствам команды, перехватывать прерывания и обрабатывать ошибки; она также должна обеспечивать интерфейс между устройствами и остальной частью системы.

Основная идея организации программного обеспечения ввода-вывода состоит в разбиении его на несколько уровней, причем нижние уровни скрывают особенности аппаратуры от верхних уровней, а те, в свою очередь, обеспечивают удобный интерфейс для пользователей.

Кроме управления вводом-выводом на уровне локальной ОС, СОС также должна уметь работать с сетевыми устройствами, потоком данных от сети к программам СОС, реализовывать множество стеков протоколов с возможностью добавления поддержки новых, без перекомпиляции ядра. Поэтому сетевые функции выносят за пределы микроядер.

Файловая система.

Файловая система локальной ОС - часть операционной системы, назначение которой состоит в том, чтобы обеспечить пользователю удобный интерфейс при работе с данными, хранящимися на диске, и обеспечить совместное использование файлов несколькими пользователями и процессами.

В микроядерной архитектуре файловая система перестает быть частью ОС, что соответствует тенденция современного компьютерного рынка, когда файловые системы начинают представляться как отдельные программные продукты.

Разработчики новых операционных систем стремятся обеспечить пользователя возможностью работать сразу с несколькими файловыми системами.

Современна файловая система имеет многоуровневую структуру, на верхнем уровне которой располагается так называемый переключатель файловых систем (в Windows 95, например, такой переключатель называется устанавливаемым диспетчером файловой системы - installable filesystem manager, IFS). Он обеспечивает интерфейс между запросами приложения и конкретной файловой системой, к которой обращается это приложение. Переключатель файловых систем преобразует запросы в формат, воспринимаемый уровнем файловых систем.

Для выполнения своих функций драйверы файловых систем обращаются к подсистеме ввода-вывода, которая отвечает за загрузку, инициализацию и управление всеми модулями низших уровней файловой системы, т.е. просмотр регистров контроллера, ???

Современная файловая система должна поддерживать возможность работы с дисковыми массивами RAID.

Ключевым компонентом любой СОС является поддержка распределенной файловой системой. Файловая система поддерживается одной или более машинами, называемыми файл-серверами. Файл-серверы перехватывают запросы на чтение или запись файлов, поступающие от других машин (не

серверов). Эти другие машины называются клиентами. Каждый посланный запрос проверяется и выполняется, а ответ отсылается обратно. Файл-серверы обычно содержат иерархические файловые системы, каждая из которых имеет корневой каталог и каталоги более низких уровней. Рабочая станция может подсоединять и монтировать эти файловые системы к своим локальным файловым системам. При этом монтируемые файловые системы остаются на серверах.

Также распределенная файловая система должна гарантировать безопасность и надежность не только во время хранения, но и во времяпередачи данных. База данных службы NDS представляет собой многоуровневую базу данных объектов, поддерживающую информацию о ресурсах всех серверов сети.

Характеристики:

• распределенность
• прозрачность
• глобальность

Распределенность заключается в том, что информация не хранится на одном сервере, а разделена на части, называемые разделами.

Прозрачность заключается в том, что NDS автоматически создает связи между программными и аппаратными компонентами, которые обеспечивают пользователю доступ к сетевым ресурсам. NDS при этом не требует от пользователя знаний физического расположения этих ресурсов. Задавая сетевой ресурс по имени, вы получите к нему корректный доступ даже в случае изменения его сетевого адреса или места расположения.

Глобальность NDS заключается в том, что после входа вы получаете доступ к ресурсам всей сети. Вместо входа в отдельный сервер пользователь NDS входит в сеть. После чего он получает доступ к разрешенным для него ресурсам сети. Информация, предоставляемая во время логического входа, используется для процесса идентификации пользователя. Позже, при попытке пользователя получить доступ к ресурсам, таким как серверы, тома или принтеры, фоновый процесс идентификации проверяет, имеет ли пользователь право использовать данный ресурс.

Таким образом, при построении современных операционных систем все чаще разработчики используют микроядерный подход. Системы управления отдельными ресурсами отделяются от ОС, а последняя, с учетом микроядерного подхода постепенно переходит от управления конкретными устройствами к управлению абстрактными ресурсами. СОС движутся в сторону универсализации, все чаще они предоставляют возможность сторонним производителям создавать собственные модули для встраивания в структуру системы, что позволяет реализовывать файловые системы, службы поддержки каталогов, службы печати и т.д. как отдельные программные продукты. Так, например, менеджер ввода/вывода ядра Windows NT н обращается к драйверам устройств напрямую, а использует для своей работы драйверы файловых систем. Значит, для поддержки новой файловой системы стороннего производителя не нужно перекомпилировать

ядро или иным образом включать в работу производителя ОС, а достаточно написать драйвер файловой системы, установить его и файловая система становится доступной в рамках ОС.

Лекция 3 Современные операционные системы

Тема: Современные сетевые операционные системы

Сетевые операционные системы:
Novell NetWare
LANtastic
Microsoft Windows (NT, XP, Vista, 7, 8)
Различные UNIX системы, такие как Solaris, FreeBSD
Различные GNU/Linux системы
IOS
ZyNOS компании ZyXEL

NetWare — сетевая операционная система и набор сетевых протоколов, которые используются в этой системе для взаимодействия с компьютерами-клиентами, подключёнными к сети. Операционная система NetWare создана компанией Novell. NetWare является закрытой операционной системой, использующей кооперативную многозадачность для выполнения различных служб на компьютерах с архитектурой Intel x86. В основе сетевых протоколов системы лежит стек протоколов Xerox Network Systems (англ.) (XNS). В настоящее время NetWare поддерживает протоколы TCP/IP и IPX/SPX. NetWare является одним из семейств XNS-систем. К таким системам, например, относятся Banyan VINES и Ungerman-Bass Net/One. В отличие от этих продуктов и XNS, система NetWare заняла существенную долю рынка в начале 1990-х и выдержала конкуренцию с Microsoft Windows NT, после выпуска которой прекратили своё существование другие конкурирующие с ней системы.
В основу NetWare была положена очень простая идея: один или несколько выделенных серверов подключаются к сети и предоставляют длясовместного использования своё дисковое пространство в виде «томов». На компьютерах-клиентах с операционной системой MS-DOS запускается несколько специальных резидентных программ, которые позволяют
«назначать» буквы дисков на тома. Пользователям необходимо зарегистрироваться в сети, чтобы получить доступ к томам и иметь возможность назначать буквы дисков. Доступ к сетевым ресурсам определяется именем регистрации.
Пользователи могут также подключаться к совместно используемым принтерам на выделенном сервере и выполнять печать на сетевых принтерах так же, как и на локальных.

Несмотря на то, что в ранних версиях NetWare все модули системы считались ненадёжными (любой неправильно работающий модуль мог нарушить работу всей системы), она была очень стабильной системой. Нередки случаи, когда серверы NetWare работают без вмешательства человека годами.

LANtastic — сетевая операционная система для DOS, Windows, Novell NetWare и OS/2. LANtastic поддерживает технологии Ethernet, ARCNET и Token Ring, а также её собственные адаптеры витой пары на 2 Мбит/с.
Её многоплатформенная поддержка позволяет станции LANtastic подключаться к любой комбинации Windows или операционных системах DOS, и его межсвязи позволяют делиться файлами, принтерами, CD-ROM и приложениями по всем предприятиям. LANtastic был особенно популярным до того, как в Windows 95 была встроена поддержка сетей и был почти лидером на рынке операционных систем.
LANtastic была первоначально разработана Artisoft Inc. в Тусон, штат Аризона. После выхода TeleVantage, LANtastic и прочая унаследованная продукция Artisoft были приобретены Spartacom Technologies в 2000 году. Позже SpartaCom была приобретена PCMicro.
В настоящее время (2006 год) самая новая версия — LANtastic 8.01. Он может соединить компьютеры с операционной системой DOS 5.0 (или выше) с Windows 3.x или выше (включая Windows XP).
Сети LANtastic используют протокол Server Message Block (SMB). Подробности очень плохо изучены.
Для совершения сессий используется Local Session Number (LSN).

Microsoft Windows — семейство проприетарных операционных систем корпорации Microsoft, ориентированных на применении графического интерфейса при управлении. Изначально Windows была всего лишь графической надстройкой для MS-DOS.
По состоянию на май 2013 года под управлением операционных систем семейства Windows по данным ресурса Netmarketshare (Net Applications) работает около 91 % персональных компьютеров[1].
Операционные системы Windows работают на платформах x86, x86-x64, IA-64, ARM. Существовали также версии для DEC Alpha, MIPS, PowerPC и SPARC[2].

Семейство Windows NT

Операционные системы этого семейства в настоящее время работают на процессорах с архитектурами x86, x64, и Itanium, ARM. Ранние версии (до 4.0 включительно) также поддерживали некоторые RISC-процессоры: Alpha, MIPS, и Power PC. Все операционные системы этого семейства являются полностью 32- или 64- битными операционными системами, и не нуждаются в MS-DOS даже для загрузки.
Только в этом семействе представлены операционные системы для серверов. До версии Windows 2000 включительно они выпускались под тем же

названием, что и аналогичная версия для рабочих станций, но с добавлением суффикса, например, «Windows NT 4.0 Server» и «Windows 2000 Datacenter Server». Начиная с Windows Server 2003 серверные операционные системы называются по-другому.

Windows NT 3.1 (1993)
Windows NT 3.5 (1994)
Windows NT 3.51 (1995)
Windows NT 4.0 (1996)
Windows 2000 — Windows NT 5.0 (2000)
Windows XP — Windows NT 5.1 (2001)
Windows XP 64-bit Edition — Windows NT 5.2 (2003)
Windows Server 2003 — Windows NT 5.2 (2003)
Windows XP Professional x64 Edition — Windows NT 5.2 (2005)
Windows Vista — Windows NT 6.0 (2006)
Windows Home Server — Windows NT 5.2 (2007)
Windows Server 2008 — Windows NT 6.0 (2008)
Windows Small Business Server — Windows NT 6.0 (2008)
Windows 7 — Windows NT 6.1 (2009)
Windows Server 2008 R2 — Windows NT 6.1 (2009)
Windows Home Server 2011 — Windows NT 6.1 (2011)
Windows 8 — Windows NT 6.2 (2012)
Windows Server 2012 — Windows NT 6.2 (2012)
Windows 8.1 - Windows NT 6.3 (2013)
Windows Server 2012 R2 — Windows NT 6.3 (2013)

В основу семейства Windows NT положено разделение адресныхпространств между процессами. Каждый процесс имеет возможность работать с выделенной ему памятью. Однако он не имеет прав для записи в память других процессов, драйверов и системного кода.

Семейство Windows NT относится к операционным системам с вытесняющей многозадачностью. Разделение процессорного времени между потоками происходит по принципу «карусели». Ядро операционной системывыделяет квант времени (в Windows 2000 квант равен примерно 20 мс) каждому из потоков по очереди при условии, что все потоки имеют одинаковый приоритет. Поток может отказаться от выделенного ему кванта времени. В этом случае система перехватывает у него управление (даже есливыделенный квант времени не закончен) и передаёт управление другому потоку. При передаче управления другому потоку система сохраняет состояние всех регистров процессора в особой структуре в оперативной памяти. Эта структура называется контекстом потока. Сохранение контекста потока достаточно для последующего возобновления его работы.

Семейство ОС для карманных компьютеров

Это семейство операционных систем реального времени было специально разработано для мобильных устройств. Поддерживаются процессоры ARM, MIPS, SuperH и x86. В отличие от остальных операционных систем

Windows, операционные системы этого семейства продаются только в составе готовых устройств, таких как смартфоны, карманные компьютеры, GPS-навигаторы, MP3-проигрыватели и другие. В настоящее время под термином «Windows CE» понимают только ядро операционной системы. Например, Windows Mobile 5.0 включает в себя ядро Windows CE 5.0.
Windows CE
Windows Mobile
Windows Phone

Семейство встраиваемых ОС Windows Embedded

Windows Embedded — это семейство операционных систем реального времени, было специально разработано для применения в различных встраиваемых системах. Ядро системы имеет общее с семейством ОС Windows CE и поддерживает процессоры ARM, MIPS, SuperH и x86. Windows Embedded включает дополнительные функции по встраиванию, среди которых фильтр защиты от записи (EWF и FBWF), загрузка с флеш-памяти, CD-ROM, сети, использование собственной оболочки системы и т. п. В отличие от операционных систем Windows, операционные системы этого семейства продаются только в составе готовых устройств, таких как: банкоматы, медицинские приборы, навигационное оборудование, «тонкие» клиенты, VoIP-терминалы, медиапроигрыватели, цифровые рамки (альбомы), кассовые терминалы, платёжные терминалы, роботы, игровые автоматы, музыкальные автоматы и другие.
В настоящее время выпускаются следующие варианты ОС Windows Embedded[8]:
Windows Embedded CE,
Windows Embedded Standard,
Windows Embedded POSReady,
Windows Embedded Enterprise,
Windows Embedded NavReady,
Windows Embedded Server.

Linux — общее название Unix-подобных операционных систем, основанных на одноимённом ядре. Ядро Linux и обычно использующиеся вместе с ним компоненты создаются и распространяются в соответствии с моделью разработки свободного и открытого программного обеспечения. Поэтому общее название не подразумевает какой-либо единой «официальной» комплектации Linux; они распространяются в основном бесплатно в виде различных готовых дистрибутивов, имеющих свой набор прикладных программ и уже настроенных под конкретные нужды пользователя.
На начальном этапе Linux бесплатно разрабатывался только энтузиастами-добровольцами, но с успехом Linux и его массовым коммерческим использованием дорабатывать ОС и вносить свой вклад стали и компании, со временем став значительной силой. Подавляющее большинство ПО в

современных дистрибутивах по-прежнему доступно по свободным лицензиям, как правило за исключением небольшого количества проприетарных компонентов. В 2008 году расчёты показывали, что для того чтобы «с нуля» разработать систему, аналогичную Fedora 9, потребовалось бы затратить 10,8 млрд долл.[6] Совокупная себестоимость ядра Linux оценена в более чем 1 млрд евро (около 1,4 млрд долл.). Только за 2008 год себестоимость ядра Linux увеличилась на 225 млн евро. В системе Linux воплощён труд в эквиваленте 73 тыс. человеко-лет.

В настоящее время системы Linux лидируют на рынках смартфонов (Android занимает 64,1 % рынка), интернет-серверов (60 %), самых мощных суперкомпьютеров (93,8 %), а также, согласно Linux Foundation, в дата-центрах и на предприятиях, занимают половину рынка встраиваемых систем, имеют значительную долю рынка нетбуков (32 % на 2009 год). На рынке домашних компьютеров Linux прочно занимает 3 место (по разным данным, от 1 до 5 %). Согласно исследованию Goldman Sachs, в целом, рыночная доля Linux среди электронных устройств составляет около 42 %.

С тех пор, как ядро Linux было создано для x86-ПК, оно было портировано на множество платформ, включая x86-64, PowerPC и ARM. Linux работает в роутерах, телевизорах и игровых приставках. ОС на ядре продолжают быстро совершенствоваться (например, новая версия ядра выпускается каждые 2-3 месяца, с 2005 года в разработке ядра принимают участие более 7800 разработчиков из более чем 800 различных компаний) и набирать популярность (за 9 месяцев с мая 2011 по январь 2012 доля Linux выросла на 64 %).

Наиболее популярными дистрибутивами являются: deb-based (Debian, Mint, Ubuntu), RPM-based (RedHat, Fedora, Mageia, OpenSUSE), source-based (Slackware, Gentoo), pacman-based Arch Linux.

Собственные дистрибутивы Linux выпускаются различными компаниями и энтузиастами со всего мира, в том числе, из России и Украины.

Solaris — компьютерная операционная система, разработанная компанией Sun Microsystems, которая ныне принадлежит Oracle Corporation. Несмотря на то что Solaris — операционная система с закрытым исходным кодом, бо льшая его часть открыта и опубликована в проекте OpenSolaris.

FreeBSD — свободная Unix-подобная операционная система, потомок AT&T Unix по линии BSD, созданной в университете Беркли. FreeBSD работает на PC-совместимых системах семейства x86, включая Microsoft Xbox, а также на DEC Alpha, Sun UltraSPARC, IA-64, AMD64, PowerPC, NEC PC-98, ARM. Готовится поддержка архитектуры MIPS.

FreeBSD разрабатывается как целостная операционная система. Исходный код ядра, драйверов устройств и базовых пользовательских программ (т. н. userland), таких как командные оболочки и т. п., содержится в одном дереве системы управления версиями (до 31 мая 2008 — CVS, сейчас — SVN). Это отличает FreeBSD от GNU/Linux — другой свободной UNIX-подобной

операционной системы — в которой ядро разрабатывается одной группой разработчиков, а набор пользовательских программ — другими (например, проект GNU), а многочисленные группы собирают это всё в единое целое и выпускают в виде различных дистрибутивов Linux.
FreeBSD хорошо зарекомендовала себя как система для построения интранет и интернет-сетей и серверов. Она предоставляет надёжные сетевые службы и эффективное управление памятью.
Помимо своей стабильности, FreeBSD популярна и благодаря своей лицензии, которая существенно отличается от широко известной лицензии GNU GPL — она позволяет использовать код не только в свободном ПО, но и в проприетарном. В отличие от GNU LGPL, которая тоже позволяет использовать свободный код в закрытой программе[3], лицензия BSD более простая и короткая.

Cisco IOS (от англ. Internetwork Operating System — Межсетевая Операционная Система) — программное обеспечение, используемое в маршрутизаторах Cisco и некоторых сетевых коммутаторах. Cisco IOS — многозадачная операционная система, выполняющая функции сетевой организации, маршрутизации, коммутации и передачи данных.
Cisco IOS имеет специфичный интерфейс командной строки (command line interface, CLI), который был скопирован многими другими сетевыми продуктами. Интерфейс IOS имеет набор многословных команд, доступные команды определены «режимом» и уровнем привилегий данного пользователя. «Global configuration mode» — дает возможность для изменения системных настроек и настроек сетевого интерфейса. Вот, например, типичные команды:
r**outer ospf 100**
aaa authorization exec default local group radius
no ip cef traffic-statistics
Всем командам приписывается определённый уровень привилегий от 0 до 15, и к ним могут обратиться только пользователи с данным уровнем привилегий. Через командный интерфейс можно определить доступные команды для каждого уровня привилегий.

ZyNOS – это сетевая операционная система, являющаяся собственностью корпорации ZyXEL. Она является платформой всех маршрутизаторов Prestige, которая обеспечивает сетевые сервисы и приложения. Она построена модульным способом, поэтому разработчикам легко добавлять в неё новые функции.

Лекция 4 Программное обеспечение виртуальных частных сетей (VPN)

Тема: Программное обеспечение виртуальных частных сетей (VPN): Структура VPN. Классификация VPN.

Введение

В последнее время в мире телекоммуникаций наблюдается повышенный интерес к виртуальным частным сетям (Virtual Private Network - VPN). Это обусловлено необходимостью снижения расходов на содержание корпоративных сетей за счет более дешевого подключения удаленных офисов и удаленных пользователей через сеть Internet. Действительно, при сравнении стоимости услуг по соединению нескольких сетей через Internet, например, с сетями Frame Relay можно заметить существенную разницу в стоимости. Однако необходимо отметить, что при объединении сетей через Internet, сразу же возникает вопрос о безопасности передачи данных, поэтому возникла необходимость создания механизмов позволяющих обеспечить конфиденциальность и целостность передаваемой информации. Сети, построенные на базе таких механизмов, и получили название VPN.

Кроме того, очень часто современному человеку, развивая свой бизнес, приходится много путешествовать. Это могут быть поездки в отдаленные уголки нашей страны или в страны зарубежья. Нередко людям нужен доступ к своей информации, хранящейся на их домашнем компьютере, или на компьютере фирмы. Эту проблему можно решить, организовав удалённый доступ к нему с помощью модема и телефонной линии. Использование телефонной линии имеет свои особенности. Недостатки этого решения в том, что звонок с другой страны стоит немалых денег. Есть и другое решение под названием VPN. Преимущества технологии VPN в том, что организация удалённого доступа делается не через телефонную линию, а через Internet, что намного дешевле и лучше.

1. Понятие и классификация VPN сетей, их построение

1.1 Что такое VPN

VPN (англ. Virtual Private Network - виртуальная частная сеть) - логическая сеть, создаваемая поверх другой сети, например Internet. Несмотря на то, что коммуникации осуществляются по публичным сетям с использованием небезопасных протоколов, за счёт шифрования создаются закрытые от посторонних каналы обмена информацией. VPN позволяет объединить, например, несколько офисов организации в единую сеть с использованием для связи между ними неподконтрольных каналов.

По своей сути VPN обладает многими свойствами выделенной линии, однако развертывается она в пределах общедоступной сети, например Интернета. С помо-щью методики туннелирования пакеты данных транслируются через общедоступную сеть как по обычному двухточечному соединению. Между каждой парой «отправитель-получатель данных» устанавливается своеобразный туннель - безопасное логическое соединение, позволяющее инкапсулировать данные одного протокола в пакеты другого.

Основными компонентами туннеля являются:

- инициатор
- маршрутизируемая сеть;

• туннельный коммутатор;
• один или несколько туннельных терминаторов.

Сам по себе принцип работы VPN не противоречит основным сетевым технологиям и протоколам. Например, при установлении соединения удаленного доступа клиент посылает серверу поток пакетов стандартного протокола PPP. В случае организации виртуальных выделенных линий между локальными сетями их маршрутизаторы также обмениваются пакетами PPP. Тем не менее, принципиально новым моментом является пересылка пакетов через безопасный туннель, организованный в пределах общедоступной сети.

Туннелирование позволяет организовать передачу пакетов одного протокола в логической среде, использующей другой протокол. В результате появляется возможность решить проблемы взаимодействия нескольких разнотипных сетей, начиная с необходимости обеспечения целостности и конфиденциальности передаваемых данных и заканчивая преодолением несоответствий внешних протоколов или схем адресации.

Существующая сетевая инфраструктура корпорации может быть подготовлена к использованию VPN как с помощью программного, так и с помощью аппаратного обеспечения. Организацию виртуальной частной сети можно сравнить с прокладкой кабеля через глобальную сеть. Как правило, непосредственное соединение между удаленным пользователем и оконечным устройством туннеля устанавливается по протоколу PPP.

Наиболее распространенный метод создания туннелей VPN - инкапсуляция сетевых протоколов (IP, IPX, AppleTalk и т.д.) в PPP и последующая инкапсуляция образованных пакетов в протокол туннелирования. Обычно в качестве последнего выступает IP или (гораздо реже) ATM и Frame Relay. Такой подход называется туннелированием второго уровня, поскольку «пассажиром» здесь является протокол именно второго уровня.

Альтернативный подход - инкапсуляция пакетов сетевого протокола непосредственно в протокол туннелирования (например, VTP) называется туннелированием третьего уровня.

Независимо от того, какие протоколы используются или какие цели преследуются при организации туннеля, основная методика остается практически неизменной.

Обычно один протокол используется для установления соединения с удаленным узлом, а другой - для инкапсуляции данных и служебной информации с целью передачи через туннель.

1.2 Классификация VPN сетей

Классифицировать VPN решения можно по нескольким основным параметрам:

1. По типу используемой среды:

• Защищённые VPN сети. Наиболее распространённый вариант приватных частных сетей. С его помощью возможно создать надежную и защищенную подсеть на основе ненадёжной сети, как правило, Интернета. Примером защищённых VPN являются: IPSec, OpenVPN и PPTP.
• Доверительные VPN сети. Используются в случаях, когда передающую среду можно считать надёжной и необходимо решить лишь задачу создания виртуальной подсети в рамках большей сети. Вопросы обеспечения безопасности становятся неактуальными. Примерами подобных VPN решении являются: MPLS и L2TP. Корректнее сказать, что эти протоколы перекладывают задачу обеспечения безопасности на другие, например L2TP, как правило, используется в паре с IPSec.

2. По способу реализации:
• VPN сети в виде специального программно-аппаратного обеспечения. Реализация VPN сети осуществляется при помощи специального комплекса программно-аппаратных средств. Такая реализация обеспечивает высокую производительность и, как правило, высокую степень защищённости.
• VPN сети в виде программного решения. Используют персональный компьютер со специальным программным обеспечением, обеспечивающим функциональность VPN.
• VPN сети с интегрированным решением. Функциональность VPN обеспечивает комплекс, решающий также задачи фильтрации сетевого трафика, организации сетевого экрана и обеспечения качества обслуживания.

3. По назначению:
• Intranet VPN. Используют для объединения в единую защищённую сеть нескольких распределённых филиалов одной организации, обменивающихся данными по открытым каналам связи.
• Remote Access VPN. Используют для создания защищённого канала между сегментом корпоративной сети (центральным офисом или филиалом) и одиночным пользователем, который, работая дома, подключается к корпоративным ресурсам с домашнего компьютера или, находясь в командировке, подключается к корпоративным ресурсам при помощи ноутбука.
• Extranet VPN. Используют для сетей, к которым подключаются «внешние» пользователи (например, заказчики или клиенты). Уровень доверия к ним намного ниже, чем к сотрудникам компании, поэтому требуется обеспечение специальных «рубежей» защиты, предотвращающих или ограничивающих доступ последних к особо ценной, конфиденциальной информации.
4. По типу протокола:
Существуют реализации виртуальных частных сетей под TCP/IP, IPX и AppleTalk. Но на сегодняшний день наблюдается тенденция к всеобщему переходу на прото-кол TCP/IP, и абсолютное большинство VPN решений поддерживает именно его.

5. По уровню сетевого протокола:
По уровню сетевого протокола на основе сопоставления с уровнями эталонной сетевой модели ISO/OSI.

1.3. Построение VPN

Существуют различные варианты построения VPN. При выборе решения требуется учитывать факторы производительности средств построения VPN. Например, если маршрутизатор и так работает на пределе мощности своего процессора, то добавление туннелей VPN и применение шифрования / дешифрования информации могут остановить работу всей сети из-за того, что этот маршрутизатор не будет справляться с простым трафиком, не говоря уже о VPN. Опыт показывает, что для построения VPN лучше всего использовать специализированное оборудование, однако если имеется ограничение в средствах, то можно обратить внимание на чисто программное решение. Рассмотрим некоторые варианты построения VPN.

• VPN на базе брандмауэров

Брандмауэры большинства производителей поддерживают туннелирование и шифрование данных. Все подобные продукты основаны на том, что трафик, проходящий через брандмауэр шифруется. К программному обеспечению собственно брандмауэра добавляется модуль шифрования. Недостатком этого метода можно назвать зависимость производительности от аппаратного обеспечения, на котором работает брандмауэр. При использовании брандмауэров на базе ПК надо помнить, что подобное решение можно применять только для небольших сетей с небольшим объемом передаваемой информации.

• VPN на базе маршрутизаторов

Другим способом построения VPN является применение для создания защищенных каналов маршрутизаторов. Так как вся информация, исходящая из локальной сети, проходит через маршрутизатор, то целесообразно возложить на этот маршрутизатор и задачи шифрования.

Примером оборудования для построения VPN на маршрутизаторах является оборудование компании Cisco Systems. Начиная с версии программного обеспечения IOS 11.3, маршрутизаторы Cisco поддерживают протоколы L2TP и IPSec. Помимо простого шифрования проходящей информации Cisco поддерживает и другие функции VPN, такие как идентификация при установлении туннельного соединения и обмен ключами.

Для повышения производительности маршрутизатора может быть использован дополнительный модуль шифрования ESA. Кроме того, компания Cisco System выпустила специализированное устройство для VPN, которое так и называется Cisco 1720 VPN Access Router (маршрутизатор доступа к VPN), предназначенное для установки в компаниях малого и среднего размера, а также в отделениях крупных организаций.

• VPN на базе программного обеспечения

Следующим подходом к построению VPN являются чисто программные решения. При реализации такого решения используется специализированное программное обеспечение, которое работает на выделенном компьютере, и в большинстве случаев выполняет роль proxy-сервера. Компьютер с таким программным обеспечением может быть расположен за брандмауэром.

В качестве примера такого решения можно выступает программное обеспечение AltaVista Tunnel 97 компании Digital. При использовании данного программного обеспечения клиент подключается к серверу Tunnel 97, аутентифицируется на нем и обменивается ключами. Шифрация производится на базе 56 или 128 битных ключей, полученных в процессе установления соединения. Далее, зашифрованные пакеты инкапсулируются в другие IP-пакеты, которые в свою очередь отправляются на сервер. Кроме того, данное программное обеспечение каждые 30 минут генерирует новые ключи, что значительно повышает защищенность соединения.

Положительными качествами AltaVista Tunnel 97 являются простота установки и удобство управления. Минусами данной системы можно считать нестандартную архитектуру (собственный алгоритм обмена ключами) и низкую производительность.

• VPN на базе сетевой ОС

Решения на базе сетевой ОС мы рассмотрим на примере системы Windows компании Microsoft. Для создания VPN Microsoft использует протокол PPTP, который интегрирован в систему Windows. Данное решение очень привлекательно для организаций использующих Windows в качестве корпоративной операционной системы. Необходимо отметить, что стоимость такого решения значительно ниже стоимости прочих решений. В работе VPN на базе Windows используется база пользователей NT, хранящаяся на Primary Domain Controller (главный контроллер домена) (PDC). При подключении к PPTP-серверу пользователь аутентифицируется по протоколам PAP, CHAP или MS-CHAP. Передаваемые пакеты инкапсулируются в пакеты GRE/PPTP. Для шифрования пакетов используется нестандартный протокол от Microsoft Point-to-Point Encryptionc 40 или 128 битным ключом, получаемым в момент установки со-единения. Недостатками данной системы являются отсутствие проверки целостности данных и невозможность смены ключей во время соединения. Положительными моментами являются легкость интеграции с Windows и низкая стоимость.

• VPN на базе аппаратных средств

Вариант построения VPN на специальных устройствах может быть использован в сетях, требующих высокой производительности. Примером такого решения служит продукт с IPro-VPN компании Radguard. Данный продукт использует аппаратное шифрование передаваемой информации, способное пропускать поток в 100 Мбит/с. IPro-VPN поддерживает протокол

IPSec и механизм управления ключами ISAKMP/Oakley. Помимо прочего, данное устройство поддерживает средства трансляции сетевых адресов и может быть дополнено специальной платой, добавляющей функции брандмауэра.

Лекция 5 Технология построения виртуальной частной сети — протоколы IPSec, SSL.

Тема: Технология построения виртуальной частной сети — протоколы IPSec, SSL.

1. Протоколы VPN сетей

Сети VPN строятся с использованием протоколов туннелирования данных через сеть связи общего пользования Интернет, причем протоколы туннелирования обеспечивают шифрование данных и осуществляют их сквозную передачу между пользователями. Как правило, на сегодняшний день для построения сетей VPN используются протоколы следующих уровней:

- Канальный уровень
- Сетевой уровень
- Транспортный уровень.

1.1 Канальный уровень

На канальном уровне могут использоваться протоколы туннелирования данных L2TP и PPTP, которые используют авторизацию и аутентификацию.

• PPTP

Протокол VPN- протокол двухточечной туннельной связи или Point-to-Point Tunnelling Protocol - PPTP. Разработан он компаниями 3Com и Microsoft с целью предоставления безопасного удаленного доступа к корпоративным сетям через Интернет. PPTP использует существующие открытые стандарты TCP/IP и во многом полагается на устаревший протокол двухточечной связи PPP. На практике PPP так и остается коммуникационным протоколом сеанса соединения PPTP. PPTP создает туннель через сеть к NT-серверу получателя и передает по нему PPP-пакеты удаленного пользователя. Сервер и рабочая станция используют виртуальную частную сеть и не обращают внимания на то, насколько безопасной или доступной является глобальная сеть между ними. Завершение сеанса соединения по инициативе сервера, в отличие от специализированных серверов удаленного доступа, позволяет администраторам локальной сети не пропускать удаленных пользователей за пределы системы безопасности Windows NT Server.

Хотя компетенция протокола PPTP распространяется только на устройства, работающие под управлением Windows, он предоставляет компаниям

возможность взаимодействовать с существующими сетевыми инфраструктурами и не наносить вред собственной системе безопасности. Таким образом, удаленный пользователь может подключиться к Интернету с помощью местного провайдера по аналоговой телефонной линии или каналу ISDN и установить соединение с сервером NT. При этом компании не приходится тратить большие суммы на организацию и обслуживание пула модемов, предоставляющего услуги удаленного доступа.

Далее рассматривается работа PPTP. PPTP инкапсулирует пакеты IP для передачи по IP-сети. Клиенты PPTP используют порт назначения для создания управляющего туннелем соединения. Этот процесс происходит на транспортном уровне модели OSI. После создания туннеля компьютер-клиент и сервер начинают обмен служебными пакетами. В дополнение к управляющему соединению PPTP, обеспечивающему работоспособность канала, создается соединение для пересылки по туннелю данных. Инкапсуляция данных перед пересылкой через туннель происходит несколько иначе, чем при обычной передаче. Инкапсуляция данных перед отправкой в туннель включает два этапа:

1. Сначала создается информационная часть PPP. Данные проходят сверху вниз, от прикладного уровня OSI до канального.

2. Затем полученные данные отправляются вверх по модели OSI и инкапсулируются протоколами верхних уровней.

Таким образом, во время второго прохода данные достигают транспортного уровня. Однако информация не может быть отправлена по назначению, так как за это отвечает канальный уровень OSI. Поэтому PPTP шифрует поле полезной нагрузки пакета и берет на себя функции второго уровня, обычно принадлежащие PPP, т.е. добавляет к PPTP-пакету PPP-заголовок и окончание. На этом создание кадра канального уровня заканчивается.
Далее, PPTP инкапсулирует PPP-кадр в пакет Generic Routing Encapsulation (GRE), который принадлежит сетевому уровню. GRE инкапсулирует протоколы сетевого уровня, например IPX, AppleTalk, DECnet, чтобы обеспечить возможность их передачи по IP-сетям. Однако GRE не имеет возможности устанавливать сессии и обеспечивать защиту данных от злоумышленников. Для этого используется способность PPTP создавать соединение для управления туннелем. Применение GRE в качестве метода инкапсуляции ограничивает поле действия PPTP только сетями IP.
После того как кадр PPP был инкапсулирован в кадр с заголовком GRE, выполняется инкапсуляция в кадр с IP-заголовком. IP-заголовок содержит адреса отправителя и получателя пакета. В заключение PPTP добавляет PPP заголовок и окончание. На приложении 3 показана структура данных для пересылки по туннелю PPTP.

Система-отправитель посылает данные через туннель. Система-получатель удаляет все служебные заголовки, оставляя только данные PPP.

• L2TP

В ближайшем будущем ожидается рост количества виртуальных частных сетей, развернутых на базе нового протокола туннелирования второго уровня Layer 2 Tunneling Protocol - L2TP.

L2TP появился в результате объединения протоколов PPTP и L2F (Layer 2 Forwarding). PPTP позволяет передавать через туннель пакеты PPP, а L2F-пакеты SLIP и PPP. Во избежание путаницы и проблем взаимодействия систем на рынке телекоммуникаций, комитет Internet Engineering Task Force (IETF) рекомендовал компании Cisco Systems объединить PPTP и L2F. По общему мнению, протокол L2TP вобрал в себя лучшие черты PPTP и L2F. Главное достоинство L2TP в том, что этот протокол позволяет создавать туннель не только в сетях IP, но и в таких, как ATM, X.25 и Frame Relay. К сожалению, реализация L2TP в Windows 2000 поддерживает только IP.

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

Функциональные возможности PPTP и L2TP различны. L2TP может использоваться не только в IP-сетях, служебные сообщения для создания туннеля и пересылки по нему данных используют одинаковый формат и протоколы. PPTP может применяться только в IP-сетях, и ему необходимо отдельное соединение TCP для создания и использования туннеля. L2TPповерх IPSec предлагает больше уровней безопасности, чем PPTP, и может гарантировать почти 100-процентную безопасность важных для организации данных. Особенности L2TP делают его очень перспективным протоколом для построения виртуальных сетей.

Протоколы L2TP и PPTP отличаются от протоколов туннелирования третьего уровня рядом особенностей:

1. Предоставление корпорациям возможности самостоятельно выбирать способ аутентификации пользователей и проверки их полномочий - на собственной «территории» или у провайдера Интернет-услуг. Обрабатывая туннелированные пакеты PPP, серверы корпоративной сети получают всю информацию, необходимую для идентификации пользователей.

2. Поддержка коммутации туннелей - завершения одного туннеля и инициирования другого к одному из множества потенциальных терминаторов. Коммутация туннелей позволяет, как бы продлить PPP - соединение до необходимой конечной точки.

3. Предоставление системным администраторам корпоративной сети возможности реализации стратегий назначения пользователям прав доступа непосредственно на брандмауэре и внутренних серверах. Поскольку терминаторы туннеля получают пакеты PPP со сведениями о пользователях, они в состоянии применять сформулированные администраторами стратегии безопасности к трафику отдельных пользователей. (Туннелированиетретьего уровня не позволяет различать поступающие от провайдера пакеты,поэтому фильтры стратегии безопасности приходится применять на конечных рабочих станциях и сетевых устройствах.) Кроме того, в случае использования туннельного коммутатора появляется возможность организовать «продолжение» туннеля
второго уровня для непосредственной трансляции трафика отдельных пользователей к соответствующим внутренним серверам. На такие серверы может быть возложена задача дополнительной фильтрации пакетов.
Также на канальном уровне для организации туннелей может использоваться технология MPLS.

• MPLS
От английского Multiprotocol Label Switching - мультипротокольная коммутация по меткам - механизм передачи данных, который эмулирует различные свойства сетей с коммутацией каналов поверх сетей с коммутацией пакетов. MPLS работает на уровне, который можно было бы расположить между канальным и третьим сетевым уровнями модели OSI, и поэтому его обычно называют протоколом канально-сетевого уровня. Он был разработан с целью обеспечения универсальной службы передачи данных как для клиентов сетей с коммутацией каналов, так и сетей с коммутацией пакетов. С помощью MPLS можно передавать трафик самой разной природы, такой как IP-пакеты, ATM, SONET и кадры Ethernet.
Решения по организации VPN на канальном уровне имеют достаточно ограниченную область действия, как правило, в рамках домена провайдера.

1.2 Сетевой уровень

Сетевой уровень (уровень IP). Используется протокол IPSec реализующий шифрование и конфедициальность данных, а также аутентификацию абонентов. Применение протокола IPSec позволяет реализовать полнофункциональный доступ эквивалентный физическому подключению к корпоративной сети. Для установления VPN каждый из участников должен сконфигурировать определенные параметры IPSec, т.е. каждый клиент должен иметь программное обеспечение реализующее IPSec.
• IPSec
Естественно, никакая компания не хотела бы открыто передавать в Интернет финансовую или другую конфиденциальную информацию. Каналы VPN защищены мощными алгоритмами шифрования, заложенными в стандарты протокола безопасности IPsec. IPSec или Internet Protocol Security - стандарт,

выбранный международным сообществом, группой IETF - Internet Engineering Task Force, создает основы безопасности для Интернет- протокола (IP/ Протокол IPSec обеспечивает защиту на сетевом уровне и требует поддержки стандарта IPSec только от общающихся между собой устройств по обе стороны соединения. Все остальные устройства, расположенные между ними, просто обеспечивают трафик IP-пакетов.

Способ взаимодействия лиц, использующих технологию IPSec, принято определять термином «защищенная ассоциация» - Security Association (SA). Защищенная ассоциация функционирует на основе соглашения, заключенного сторонами, которые пользуются средствами IPSec для защиты передаваемой друг другу информации. Это соглашение регулирует несколько параметров: IP-адреса отправителя и получателя, криптографический алгоритм, порядок обмена ключами, размеры ключей, срок службы ключей, алгоритм аутентификации.

IPSec - это согласованный набор открытых стандартов, имеющий ядро, которое может быть достаточно просто дополнено новыми функциями и протоколами. Ядро IPSec составляют три протокола:

• AH или Authentication Header - заголовок аутентификации - гарантирует целост-ность и аутентичность данных. Основное назначение протокола AH - он позволяет приемной стороне убедиться, что:

1. пакет был отправлен стороной, с которой установлена безопасная ассоциация;
2. содержимое пакета не было искажено в процессе его передачи по сети; пакет не является дубликатом уже полученного пакета.

Две первые функции обязательны для протокола AH, а последняя выбирается при установлении ассоциации по желанию. Для выполнения этих функций протокол AH использует специальный заголовок. Его структура рассматривается по следующей схеме:

1. В поле следующего заголовка (next header) указывается код протокола более высокого уровня, то есть протокола, сообщение которого размещено в поле данных IP-пакета.
2. В поле длины полезной нагрузки (payload length) содержится длина заголовка AH.
3. Индекс параметров безопасности (Security Parameters Index, SPI) используется для связи пакета с предусмотренной для него безопасной ассоциацией.
4. Поле порядкового номера (Sequence Number, SN) указывает на порядковый номер пакета и применяется для защиты от его ложного воспроизведения (когда третья сторона пытается повторно использовать перехваченные защищенные пакеты, отправленные реально аутентифицированным отправителем).
5. Поле данных аутентификации (authentication data), которое содержит так называемое значение проверки целостности (Integrity Check Value, ICV), используется для аутентификации и проверки целостности пакета. Это

значение, называемое также дайджестом, вычисляется с помощью одной из двух обязательно поддерживаемых протоколом AH вычислительно необратимых функций MD5 или SAH-1, но может использоваться и любая другая функция.

• ESP или Encapsulating Security Payload - инкапсуляция зашифрованных данных - шифрует передаваемые данные, обеспечивая конфиденциальность, может также поддерживать аутентификацию и целостность данных;
Протокол ESP решает две группы задач.
1. К первой относятся задачи, аналогичные задачам протокола AH, - это обеспечение аутентификации и целостности данных на основе дайджеста,
2. Ко второй - защита передаваемых данных путем их шифрования от несанкционированного просмотра.
Заголовок делится на две части, разделяемые полем данных.
1. Первая часть, называемая собственно заголовком ESP, образуется двумя полями (SPI и SN), назначение которых аналогично одноименным полям протокола AH, и размещается перед полем данных.
2. Остальные служебные поля протокола ESP, называемые концевиком ESP, расположены в конце пакета.
Два поля концевика - следующего заголовка и данных аутентификации - аналогичны полям заголовка AH. Поле данных аутентификации отсутствует, если при установлении безопасной ассоциации принято решение не использовать возможностей протокола ESP по обеспечению целостности.
Помимо этих полей концевик содержит два дополнительных поля - заполнителя и длины заполнителя.
Протоколы AH и ESP могут защищать данные в двух режимах:
1. в транспортном - передача ведется с оригинальными IP-заголовками;
2. в туннельном - исходный пакет помещается в новый IP-пакет и передача ведется с новыми заголовками.
Применение того или иного режима зависит от требований, предъявляемых к защите данных, а также от роли, которую играет в сети узел, завершающий защи-щенный канал. Так, узел может быть хостом (конечным узлом) или шлюзом (промежуточным узлом). Соответственно, имеются три схемы применения протокола IPSec:
1. хост-хост;
2. шлюз-шлюз;
3. хост-шлюз.
Возможности протоколов AH и ESP частично перекрываются: протокол AH отвечает только за обеспечение целостности и аутентификации данных, протокол ESP может шифровать данные и, кроме того, выполнять функции протокола AH (в урезанном виде). ESP может поддерживать функции шифрования и аутентификации / целостности в любых комбинациях, то есть либо всю группу функций, либо только аутентификацию / целостность, либо только шифрование.

• IKE или Internet Key Exchange - обмен ключами Интернета - решает вспомогательную задачу автоматического предоставления конечным точкам защищенного канала секретных ключей, необходимых для работы протоколов аутентификации и шифрования данных.

1.3 Транспортный уровень

На транспортном уровне используется протокол SSL/TLS или Secure Socket Layer/Transport Layer Security, реализующий шифрование и аутентификацию между транспортными уровнями приемника и передатчика. SSL/TLS может применяться для защиты трафика TCP, не может применяться для защиты трафика UDP. Для функционирования VPN на основе SSL/TLS нет необходимости в реализации специального программного обеспечения так как каждый браузер и почтовый клиент оснащены этими протоколами. В силу того, что SSL/TLS реализуется на транспортном уровне, защищенное соединение устанавливается «из-конца-в-конец».

TLS-протокол основан на Netscape SSL-протоколе версии 3.0 и состоит из двух частей - TLS Record Protocol и TLS Handshake Protocol. Различие между SSL 3.0 и TLS 1.0 незначительные.

SSL/TLS включает в себя три основных фазы: 1) Диалог между сторонами, целью которого является выбор алгоритма шифрования; 2) Обмен ключами на основе криптосистем с открытым ключом или аутентификация на основе сертификатов; 3) Передача данных, шифруемых при помощи симметричных алгоритмов шифрования.

1.4 Реализация VPN: IPSec или SSL/TLS?

Зачастую перед руководителями IT подразделений стоит вопрос: какой из протоколов выбрать для построения корпоративной сети VPN? Ответ не очевиден так как каждый из подходов имеет как плюсы, так и минусы. Постараемся провести анализ и выявить когда необходимо применять IPSec, а когда SSL/TLS. Как видно из анализа характеристик этих протоколов они не являются взаимозаменяемыми и могут функционировать как отдельно,так и параллельно, определяя функциональные особенности каждой из реализованных VPN.

Выбор протокола для построения корпоративной сети VPN можно осуществлять по следующим критериям:

• Тип доступа необходимый для пользователей сети VPN.

1. Полнофункциональное постоянное подключение к корпоративной сети. Рекомендуемый выбор - протокол IPSec.

2. Временное подключение, например, мобильного пользователя или пользователя использующего публичный компьютер, с целью получения доступа к определенным услугам, например, электронной почте или базе данных. Рекомендуемый выбор - протокол SSL/TLS, который позволяет организовать VPN для каждой отдельной услуги.

• Является ли пользователь сотрудником компании.

1. Если пользователь является сотрудником компании, устройство которым он пользуется для доступа к корпоративной сети через IPSec VPN может быть сконфигурировано некоторым определенным способом.
2. Если пользователь не является сотрудником компании к корпоративной сети которой осуществляется доступ, рекомендуется использовать SSL/TLS. Это позволит ограничить гостевой доступ только определенными услугами.
• Каков уровень безопасности корпоративной сети.
1. Высокий. Рекомендуемый выбор - протокол IPSec. Действительно, уровень безо-пасности предлагаемый IPSec гораздо выше уровня безопасности предлагаемого протоколом SSL/TLS в силу использования конфигурируемого ПО на стороне пользователя и шлюза безопасности на стороне корпоративной сети.
2. Средний. Рекомендуемый выбор - протокол SSL/TLS позволяющий осуществлять доступ с любых терминалов.
3. В зависимости от услуги - от среднего до высокого. Рекомендуемый выбор - комбинация протоколов IPSec (для услуг требующих высокий уровень безопасности) и SSL/TLS (для услуг требующих средний уровень безопасности).
• Уровень безопасности данных передаваемых пользователем.
1. Высокий, например, менеджмент компании. Рекомендуемый выбор - протокол IPSec.
2. Средний, например, партнер. Рекомендуемый выбор - протокол SSL/TLS.
В зависимости от услуги - от среднего до высокого. Рекомендуемый выбор - комбинация протоколов IPSec (для услуг требующих высокий уровень безопасности) и SSL/TLS (для услуг требующих средний уровень безопасности).
• Что важнее, быстрое развертывание VPN или масштабируемость решения в бу-дущем.
1. Быстрое развертывание сети VPN с минимальными затратами. Рекомендуемый выбор - протокол SSL/TLS. В этом случае нет необходимости реализации специального ПО на стороне пользователя как в случае IPSec.
2. Масштабируемость сети VPN - добавление доступа к различным услугам. Реко-мендуемый выбор - протокол IPSec позволяющий осуществление доступа ко всем услугам и ресурсам корпоративной сети.
3. Быстрое развертывание и масштабируемость. Рекомендуемый выбор - комбинация IPSec и SSL/TLS: использование SSL/TLS на первом этапе для осуществления доступа к необходимым услугам с последующим внедрением IPSec.

2. Методы реализации VPN сетей

Виртуальная частная сеть базируется на трех методах реализации:
• Туннелирование;
• Шифрование;
• Аутентификация.

2.1 Туннелирование

Туннелирование обеспечивает передачу данных между двумя точками - окончаниями туннеля - таким образом, что для источника и приемника данных оказывается скрытой вся сетевая инфраструктура, лежащая между ними.

Транспортная среда туннеля, как паром, подхватывает пакеты используемого сетевого протокола у входа в туннель и без изменений доставляет их квыходу. По-строения туннеля достаточно для того, чтобы соединить двасетевых узла так, что с точки зрения работающего на них программного обеспечения они выглядят подключенными к одной (локальной) сети. Однако нельзя забывать, что на самом деле «паром» с данными проходит через множество промежуточных узлов (маршрутизаторов) открытой публичной сети.

Такое положение дел таит в себе две проблемы. Первая заключается в том, что передаваемая через туннель информация может быть перехвачена злоумышленниками. Если она конфиденциальна (номера банковских карточек, финансовые отчеты, сведения личного характера), то вполне реальна угроза ее компрометации, что уже само по себе неприятно. Хуже того, злоумышленники имеют возможность модифицировать передаваемые через туннель данные так, что получатель не сможет проверить ихдостоверность. Последствия могут быть самыми плачевными. Учитывая сказанное, мы приходим к выводу, что туннель в чистом виде пригоден разве что для некоторых типов сетевых компьютерных игр и не может претендовать на более серьезное применение. Обе проблемы решаются современными средствами криптографической защиты информации. Чтобы воспрепятствовать внесению несанкционированных изменений в пакет с данными на пути его следования по туннелю, используется метод электронной цифровой подписи (ЭЦП). Суть метода состоит в том, что каждый передаваемый пакет снабжается дополнительным блоком информации, который вырабатывается в соответствии с асимметричным криптографическим алгоритмом и уникален для содержимого пакета и секретного ключа ЭЦП отправителя. Этот блок информации является ЭЦП пакета и позволяет выполнить аутентификацию данных получателем, которому известен открытый ключ ЭЦП отправителя. Защита передаваемых через туннель данных от несанкционированного просмотра достигается путем использования сильных алгоритмов шифрования.

2.2 Аутентификация

Обеспечение безопасности является основной функцией VPN. Все данные от компьютеров-клиентов проходят через Internet к VPN-серверу. Такой сервер может находиться на большом расстоянии от клиентского компьютера, и данные на пути к сети организации проходят через оборудование множества провайдеров. Как убедиться, что данные не были прочитаны или изменены? Для этого применяются различные методы аутентификации и шифрования.

Для аутентификации пользователей PPTP может задействовать любой из протоколов, применяемых для PPP

• EAP или Extensible Authentication Protocol;
• MSCHAP или Microsoft Challenge Handshake Authentication Protocol (версии 1 и 2);
• CHAP или Challenge Handshake Authentication Protocol;
• SPAP или Shiva Password Authentication Protocol;
• PAP или Password Authentication Protocol.

Лучшими считаются протоколы MSCHAP версии 2 и Transport Layer Security (EAP-TLS), поскольку они обеспечивают взаимную аутентификацию, т.е. VPN-сервер и клиент идентифицируют друг друга. Во всех остальных протоколах только сервер проводит аутентификацию клиентов.

Хотя PPTP обеспечивает достаточную степень безопасности, но все же L2TP поверх IPSec надежнее. L2TP поверх IPSec обеспечивает аутентификацию на уровнях «пользователь» и 'компьютер', а также выполняет аутентификацию и шифрование данных.

Аутентификация осуществляется либо отрытым тестом (clear text password), либо по схеме запрос / отклик (challenge/response). С прямым текстом все ясно. Клиент посылает серверу пароль. Сервер сравнивает это с эталоном и либо запрещает доступ, либо говорит «добро пожаловать». Открытая аутентификация практически не встречается.

Схема запрос / отклик намного более продвинута. В общем виде она выглядит так:

• клиент посылает серверу запрос (request) на аутентификацию;
• сервер возвращает случайный отклик (challenge);
• клиент снимает со своего пароля хеш (хешем называется результат хеш-функции, которая преобразовывает входной массив данных произвольной длины в выходную битовую строку фиксированной длины), шифрует им отклик и передает его серверу;
• то же самое проделывает и сервер, сравнивая полученный результат с ответом клиента;
• если зашифрованный отклик совпадает, аутентификация считается успешной;

На первом этапе аутентификации клиентов и серверов VPN, L2TP поверх IPSec использует локальные сертификаты, полученные от службы сертификации. Клиент и сервер обмениваются сертификатами и создают защищенное соединение ESP SA (security association). После того как L2TP (поверх IPSec) завершает процесс аутентификации компьютера, выполняется аутентификация на уровне пользователя. Для аутентификации можно задействовать любой протокол, даже PAP, передающий имя пользователя и пароль в открытом виде. Это вполне безопасно, так как L2TP поверх IPSec шифрует всю сессию. Однако проведение аутентификации пользователя при помощи MSCHAP, применяющего различные ключи шифрования для аутентификации компьютера и пользователя, может усилить защиту.

2.3. Шифрование

Шифрование с помощью PPTP гарантирует, что никто не сможет получить доступ к данным при пересылке через Internet. В настоящее время поддерживаются два метода шифрования:

• Протокол шифрования MPPE или Microsoft Point-to-Point Encryption совместим только с MSCHAP (версии 1 и 2);

• EAP-TLS и умеет автоматически выбирать длину ключа шифрования при согласовании параметров между клиентом и сервером.

MPPE поддерживает работу с ключами длиной 40, 56 или 128 бит. Старые операционные системы Windows поддерживают шифрование с длиной ключа только 40 бит, поэтому в смешанной среде Windows следует выбирать минимальную длину ключа.

PPTP изменяет значение ключа шифрации после каждого принятого пакета. Протокол MMPE разрабатывался для каналов связи точка-точка, в которых пакеты передаются последовательно, и потеря данных очень мала. В этой ситуации значение ключа для очередного пакета зависит от результатов дешифрации предыдущего пакета. При построении виртуальных сетей через сети общего доступа эти условия соблюдать невозможно, так как пакеты данных часто приходят к получателю не в той последовательности, в какой были отправлены. Поэтому PPTP использует для изменения ключа шифрования порядковые номера пакетов. Это позволяет выполнять дешифрацию независимо от предыдущих принятых пакетов.

Оба протокола реализованы как в Microsoft Windows, так и вне ее (например, в BSD), на алгоритмы работы VPN могут существенно отличаться. В NT (и производных от нее системах). Основные сведения приведены в таблице. [Приложение 6]

Таким образом, связка «туннелирование + аутентификация + шифрование» позволяет передавать данные между двумя точками через сеть общего пользования, моделируя работу частной (локальной) сети. Иными словами, рассмотренные средства позволяют построить виртуальную частную сеть.

Дополнительным приятным эффектом VPN-соединения является возможность (и даже необходимость) использования системы адресации, принятой в локальной сети.

Реализация виртуальной частной сети на практике выглядит следующим образом. В локальной вычислительной сети офиса фирмы устанавливается сервер VPN. Удаленный пользователь (или маршрутизатор, если осуществляется соединение двух офисов) с использованием клиентского программного обеспечения VPN инициирует процедуру соединения с сервером. Происходит аутентификация пользователя - первая фаза установления VPN-соединения. В случае подтверждения полномочий наступает вторая фаза - между клиентом и сервером выполняется согласование деталей обеспечения безопасности соединения. После этого организуется VPN-соединение, обеспечивающее обмен информацией между клиентом и сервером в форме, когда каждый пакет с данными проходит

через процедуры шифрования / дешифрования и проверки целостности - аутентификации данных.
Основной проблемой сетей VPN является отсутствие устоявшихся стандартов аутентификации и обмена шифрованной информацией. Эти стандарты все еще находятся в процессе разработки и потому продукты различных производителей не могут устанавливать VPN-соединения и автоматически обмениваться ключами. Данная проблема влечет за собой замедление распространения VPN, так как трудно заставить различные компании пользоваться продукцией одного производителя, а потому затруднен процесс объединения сетей компаний-партнеров в, так называемые, extranet-сети.