# ВВЕДЕНИЕ

## 1. СООБЩЕСТВО ИНТЕРНЕТА

Сообщество Интернета (Internet Society, ISOC) — это груша профессионалов и экспертов, координирующая его жизнедеятельность и развитие, Основная деятельность ISOC — разработка стандартов и протоколов совместно с другими организациям.

Существует целый ряд некоммерческих организаций, которые обеспечивают функционирование Интернета, разрабатывая стандарты и способствуя достижению консенсуса. Среди них следует отметить:

* ·Интернет-общество (головная организация Интернета)
* ·Совет по архитектуре Интернета (*IAB*) (Стандарты зарубежной технологии)
* ·Рабочая инженерная группа по Интернету (*IETF*) (Совершенствование уровня технологий)
* ·Рабочая исследовательская группа по Интернету (*IRTF*) (Исследования в области будущего Интернета)
* ·Корпорация Интернет по выделенным именам и номерам (*ICANN*) (Заведует системой названия доменов и присвоения протокольных номеров в Интернете)
* ·*VeriSign* (в прошлом *Network Solutions*) (первый регистратор доменов и по сей день менеджер центральной базы данных) и аккредитованные регистраторы.

***IAB***

*IАВ* (Internet Architecture Board, ранее — Internet Activities Board) — техническая консультативная группа и составе *ISOC*. Она наблюдает за архитектурой и развитием протоколов Интернета, создает стандарты, управляет серией документов *RFC* (Request for Comments) и готовит различные периодические издания. Также *IАВ* сотрудничает с другими организациями, занимающимися техническими вопросами и стандартами, связанными с Интернетом. В составе *IАВ* есть две основные подчиненные группы — *IETF* (Internet Engineering Task Force) и *IRTF* (Internet Research Task Force).

***IETF и IESG***

*IETF* — Рабочая группа инженеров Интернета, разрабатывает прото­колы. Она состоит из отдельных рабочих групп, которые обновляют существующие стандарты и создают новые. Члены *IETF* обмениваются материалами с *IRTF* и рекомендуют стандарты для *IESG* (Internet Engi­neering Steering Group) — Группы управления инженерами Интернета, действующей совместно с *IAB*.

В *IFTF* разрабатываются девять направлений: приложения, межсетевые службы, управление сетями, функциональные требования, маршрутизация, безопасность, служебные приложения, транспорты и услуги пользователям, Для каждого создана одна или несколько рабочих групп, ответственных за создание стандарта или разрешение проблемы. Рабочую группу обычно утверждают на неформальном заседании *BOF* (Birds of Feather), проводимом на съезде *IETF*.

*IETF* собирается три раза в год для обсуждения оперативных и технических проблем Интернета. Если проблемы заслуживают особого внимания, *IETF* создает специальную рабочую группу. Рабочая группа подготавливает доклад или рекомендации, которые либо могут быть добровольно приняты *IETF*, либо направлены *IAB* на предмет утверждения их в качестве стандартов.

Один из ключевых технических стандартов, над которым в настоящее время работает *IETF* – это следующее поколение Интернет-протокола (IP).

***IRTF и IRSG***

*IRTF* — Исследовательская группа Интернета — проблемное подразделение *IAB*. В нее включе ны группы, которые занимаются протоколами, приложениями, архитектурой и технологиями Интернета. Ею руководят председатель *IRTF* и *IRSG* (Internet Research Steering Group) — Группа управления исследованиями Интернета, в которую входят руководители различных отделом и другие ученые.

Обязанности *IETF* и *IRTF* па первый взгляд похожи, однако *IRTF* проводит более долгосрочные исследования (людей меньше, структура неформальна, но исследования глубже), тогда как *IETF* решает краткосрочные задачи создания Стандартов (больше людей и бюрократических операции, необходимых для формализации стандартов).

***ICANN***

ICANN (Internet Corporation for Assigned Names and Number) является некоммерческой организацией, которая была создана 18 сентября 1998 года при участии правительства США. Изначально организация была создана для регулирования вопросов, которые связаны с доменными именами, IP-адресами и прочими аспектами функционирования Интернета.

Организация ICANN использует распределенную систему регистрации доменов, основанную на принципе свободного доступа аккредитованных регистраторов к реестрам доменных имен.

Все решения, которые принимаются организацией предварительно обсуждаются с представителями интернет-сообщества, бизнеса и государственных органов власти различных стран.

***VeriSign, Inc.***

VeriSign, Inc. — американская компания, поддерживающая домены первого уровня .com, .net и некоторые другие. VeriSign также поддерживает два из корневых серверов DNS: A и J.

Первая в мире компания, которая начала продавать цифровые сертификаты SSL.

***Другие организации***

***InterNIC***

*InterNIC* (Internet Network Information Center) — Центр сетевой информации Интернета, регистрирует имена доменов Интернета и управляет базой данных этих имей. Этот процесс сейчас пытаются модернизировать. Одно из возможных решений — создание дополнительных центров регистрации. Кроме того, еще не завершены несколько судебных процессов по вопросам допустимости сбора оплаты за регистрацию и назначения полномочной организации по сбору этой оплаты.

***IANA***

*IANA* (Internal Assigned Numbers Authority) контролирует распределе­ние в Интернете числовых параметров протокола IP, гарантируя, что каждый домен получает уникальное значение. Помимо IP-адресов, *IANA* служит центральным реестром для других чисел и данных, относящих­ся к Интернету. *IANA* находится в Институте информационных паук (Information Sciences Institute) при Университете Южной Калифорнии (University of Southern California**).**

***Консорциум W3C***

Тим Борнерс-Ли (Tim Berners-Lee) предложил модель WWW, будучи сотрудником Европейской лаборатории финики частиц (*CERN*), Сейчас он работает о Массачусетсом технологическом институте (Massachusetts Institute of Technology, *MIT*) и занимает пост директора консорциума *W3C*.

World Wide Web Consortium (*W3C*) — это международный, не зависящий от производителей промышленный консорциум, созданный на коммерческой основе и сотрудничающий с производителями и создателями стандартов с целью развитии Web-протоколов Интернета, таких, как *HTTP, HTML* и *URL*.

В США консорциум *W3C* существует на базе Лаборатории компьютерных наук Массачусетского технологического института (*МIТ* Laboratory for Computer Science, *MIT* *LCS*), в Европе — на базе Национального исследовательского института информатики и автоматизации (Institut National de Recherche en Informatique et en Automatique, *INRIA*) и в Азии — на базе Keio University Shonan Fujisawa.

Стандарты, связанные с Интернетом, разрабатываются и другими организациями, такими как Руководящая инженерная комиссия Интернет (IESG), Совет по архитектуре Интернета, Форум асинхронной передачи (*ATM*) и Консорциум Всемирной паутины (*W3C*).

***Некоммерческие компании***

Помимо *ISOC* и взаимосвязанных с ним организации, на развитие Интернета и других отраслей компьютерной индустрии оказывают влияние многие организации.

* Международная организация стандартизации (International Organization for Standardization, *ISO*),
* Американский национальный институт стандартов (American National Standards Institute, *ANSI*),
* Европейская ассоциация производителей компьютеров (European Computer Manufacturers Association, *ECMA*),
* Институт инженеров no электротехнике и электронике (Institute of Electrical and Electronics Engineers, *IEEE*),
* Фонд открытого программного обеспечения (Open Software Foundation, O*SF*) и другие.

Помимо прочего, в *ISO* входит целый ряд рабочих групп, которые ведут исследования по другим направлениям, например описывают форматы бумаги и типы фотопленок, хотя сама *ISO* не занимается электротехническими и инженерными аспектами исследуемых проблем.

***Коммерческие компании***

Кроме стандартизирующих организации, занимающихся разработкой протоколов Интернета, значительное влияние оказывают корпорации, использующие Интернет в коммерческих целях. Так, Microsoft, Netscape и Sun Microsystems принимают значительное участие в развитии протоколов и интерфейсов Интернета. Иногда эти интерфейсы становятся Стандартами де-факто, иногда — не получают распространения, а иногда передаются стандартизирующим организациям.

Например, Microsoft передала свои технологии ActiveX в ведение Active Group (подразделение Open Group), Теперь их дальнейшей судьбой занимается комитет из двенадцати производителей. Компания Netscape сотрудничала с *ЕСМА* для создания стандартизированной версии Java­Script, названной EcmaScript. Sun Microsystems в настоящее время работает над принятием Java и качестве стандарта *ISO*.

## 2. МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ОТКРЫТЫХ СИСТЕМ

Рассмотрим семиуровневую модель взаимодействия открытых систем (Open System Interconnection, OSI), предложенную Международ­ной организацией па стандартизации (International Organization for Standardization, ISO). Модель ESO/OSI предполагает, что все сетевые при­ложения можно подразделить на семь уровней, для каждого из которых созданы свои стандарты и общие модели. В результате задача сетевого взаимодействия делится на меньшие и более легкие задачи, обеспечи­вается совместимость между продуктами равных производителей и упрощается разработка приложений за счет создания отдельных уров­ней и использования уже существующих реализаций.

Семиуровневая модель показана на рис. 1-1.



Рис. 1-1. Семиуровневая модель ISO/OSI.



Рис. 1-2. Модель OSI и типы протоколов

Теоретически, каждым уровень должен взаимодействовать с аналогичным уровнем удаленного компьютера. На практике каждый из них, за исключением физического, взаимодействует с выше- и нижележащими уровнями — предоставляет услуги вышележащему и пользуется услуга­ми нижележащего. В реальной ситуации наодном компьютере независимо друг от друга иногда выполняется несколько реализаций одного уровня. Например, компьютер может иметь несколько сетевых адаптеров стандарта Ethernet или адаптеры стандартов Token-Ring и т. д.

В идеале архитектура ISO/OSI должна была появиться первой, а уже потом — все остальные разработки, созданные коммерческими, иссле­довательскими и стандартизирующими организациями, причем каждая — для конкретного уровня архитектуры. На самом же деле многие технологии появились раньше семиуровневой модели. Кроме того, не­которые новые технологии, разработанные позже модели ISO/OSI, не полностью с ней совместимы. Но, несмотря на столь сложный путь развития, эта модель обеспечивает достаточную совместимость, а значит, она выгодна.

***Физический уровень***

Физический уровень описывает физические свойства (например, элек­тромеханические характеристики) среды и сигналов, переносящих ин­формацию. Это физические характеристики кабелей и разъемов, уров­ни напряжения и электрического сопротивления и т. д., в том числе, например, спецификации кабеля «неэкранированная витая пара». Подробное описание физического уровня выхо­дит за рамки данной дисциплины.

***Канальный уровень***

Канальный уровень обеспечивает перенос данных по физической сре­де. Он поделен на два подуровня: управления логическим каналом (logical link control, LLC) и управления доступом к среде (media access control, МАС). Такое деление позволяет одному уровню LI..C использовать различные реализации уровни MAC. Уровень MAC работает с применяемыми в Ethernet и TokenRing физическими адресами, которые «вшиты» в сетевые адаптеры их производителями. Следует различать физические и логические (например, *IР)* адреса. С последними работает сетевой уровень.

***Сетевой уровень***

В отличие от канального уровня, имеющего дело с физическими адресами, сетевой уровень работает с логическими адресами. Он обеспечивает подключение и маршрутизацию между двумя узлами сети.

Сетевой уровень предоставляют транспортному уровню услуги с установлением логического соединения, например X.25, или при установлении оного (connectionless), например IP (Internet Protocol). Одна из основных функций сетевого уровня — маршрутизация.

К протоколам сетевого уровня относятся IP и ICMP (Internet Control Message Protocol).

***Транспортный уровень***

Транспортный уровень предоставляет услуги, аналогичные услугам се­тевого уровня. Надежность гарантируют лишь некоторые (далеко не все) реализации сетевых уровней, поэтому ее относят к числу функций, вы­полняемых транспортным уровнем. Транспортный уровень должен су­ществовать хотя бы потому, что иногда все три нижних уровня (физи­ческий, канальный и сетевой) предоставляет оператор услуг связи. В любом случае, используя соответствующий протокол транспортного уров­ня, потребитель услуг может обеспечить требуемую надежность услуг.

TCP [Transmission Control Protocol) — широко распространенный про­токол транспортного уровня.

***Сеансовый уровень***

Сеансовый уровень обеспечивает установление и разрыв сеансов, и уп­равление ими. Сеанс — это логическое соединение между двумя конеч­ными пунктами. Использование сеансового уровня всегда необходимо; например, если приложения применяют модель передачи данных для установления логического соединения, то протокол сеансового уровня абсолютно не нужен. В такой модели каждый посылаемый пакет данных содержит всю информацию о месте назначения — аналогично письму, отправляемому по почте. В модели с установлением логического соединения перед непосредственной отправкой данных выполняются мероприятия по установлению этого логического соединения (капала). После передачи данных — дополнительные действия по завершению сеанса

Сеансовый уровень следит также за очередностью передачи данных, эту функцию называют «управление диалогом».

Примеры протоколов сеансового, представительного и прикладного уровней — SMTP (Simple Mail Transfer Protocol), FTP (file Transfer Protocol) и Telnet.

***Представительный уровень***

Представительный уровень позволяет двум стекам протоколом «договориться» о синтаксисе (представлении) передаваемых друг другу данных. Поскольку гарантий одинакового представления информации нет, то этот уровень при необходимости переводит данные из одного вида в другой.

***Прикладной уровень***

Прикладной уровень— высший в модели ISO/OSI. На этом уровне выполняются конкретные приложения, которые пользуются услугами представительного уровня (и косвенно всех остальных). Это может быть обмен электронной почтой, пересылка файлов или любое другое сетевое приложение.

На рис. 1-3 показана модель ISO/OSI и некоторые протоколы соответ­ствующих уровней.



Рис. 1-3. Соответствие популярных стеков протоколов модели OSI

## 3. ОСНОВНЫЕ ТРАНСПОРТЫ И КАНАЛЫ

Перечислим основные существующие технологии передачи данных:

1. Модемы
2. Модемы на 56 кбит/с
3. Кабельные модемы
4. Ethernet
5. Gigabit Ethernet
6. TokenRing
7. FDDI
8. Линии Т1/ТЗ
9. ISDN
10. Frame relay
11. SMDS
12. ATM
13. xDSL
14. DS1/T1/E1
15. DSL/ISDN
16. HDSL
17. ADSL
18. RADSL
19. VDSL
20. SDSL
21. Беспроводные сети
22. Спутники
23. Беспроводные сети
24. WLAN-сети, Wi-Fi

***Модемы***

Модем (сокращение, от «модулятор/демодулятор») преобразует цифровые сигналы в аналоговые и наоборот и позволяет компьютерам (использующим цифровые сигналы) передавать информацию по обычным телефонным линиям (в которых применяются аналоговые сигналы). Модем-отправитель переводит цифровые сигналы компьютера в аналоговый вид, а затем отправляет их в телефонную линию. Модем-приемник выполняет обратное преобразование сигнала (из аналогового вида в цифровой) и передает его компьютеру.

Модем, инициирующий сеанс связи, пытается использовать наиболее из быстрый из доступных протоколов, а если модем-приемник не поддерживает его, то обращается к следующему протоколу.

Также модемы используют протоколы обнаружения и коррекции ошибок, которые позволяют выявить искаженные данные, а в некоторых случаях даже вычислить исходные данные без повторной их передачи. После выбора протокола модуляции аналогичным способом подключаются алгоритмы обнаружения и коррекции ошибок. Вот некоторые из них: MNP (Microcom Networking Protocol) уровнеq 1, 2, 3 и 4; V.42, также известный как LAPM (Link Access Protocol for Modems). Порядок применения этих протоколов следующий: V.42, MNP 4, MNP 3, MNP 2 и MNP I. Использование модемом протоколов обнаружения и коррекции ошибок не позволяет на более высоких уровнях совсем отказаться от применения протоколов контроля ошибок.

***Модемы на 56 кбит/с.***

Самая последняя модель модемов — те, что передают данные со скоростью 56 кбит/с по коммутируемой телефонной сети общего пользования (public switched telephone network, PSTN).

Классический модем генерирует аналоговый сигнал, который PSTN превращает в цифровой посредством дискретизации (на частоте примерно 8 000 Гц). Далее цифровой сигнал передастся па сервер или поставщику услуг Интернета (Internet service provider, ISP), при этом может произойти цифро-аналого-цифровое преобразование. Модемы же па на 56 кбит/с предполагают полностью цифровое соединение местной телефонной станции с сервером или ISP. Это проиллюстрировано на рис. 2-1.



Рис. 2-1. Подключение модема на 56 кбит/с

Скорость 56 кбит/с возможно благодаря отсутствию дополнительного аналого-цифрового преобразования при котором иногда возникают погрешности дискретизации. Модемы на 56 кбит/с асимметричны, то есть обеспечивают скорость передачи до 56 кбит/с в одном направлении и до 28 кбит/с — в другом. Это вызвано тем, что в одном направлении происходит одно аналого-цифровое преобразование, а в другом уже два.

 ***Кабельные модемы***

Кабельные модемы позволяют подключаться к Интернету через кабельные сети, так же как классические модемы — через телефонную сеть общего пользования. Скорость обычных модемов при работе в кабельной сети составляет около 28,8 кбит/с (иногда до 56 кбит/с). Большая скорость достижима потому, что используемая среда передачи (кабельная сеть) может переносить сигналы в гораздо более высоком диапазоне частот (порядка 6—8 МГц). Таким образом, кабельный модем для передачи и приема данных использует расширенный спектр частот. Кабельные модемы подключаются через трансивер адаптера Ethernet. Модем, переводит данные из формата Ethernet в аналоговый вид и наоборот.

Скорость до 30 Мбит/с — только теоретически возможна, но обычно она существенно ниже.

Причиной снижения скорости кабельных модемов заложена в архитектуре кабельных сетей, где информация проходит только водном направлении — от источника к потребителям и обратная связь не предусмотрена, использование в кабельных сетях дешевых проводов существенно снижают скорости передачи данных.

В следствии этого, реальная скорости передачи данных составляет примерно 1,5 Мбит/с.

***Ethernet***

Разработанный компанией Xerox, стандарт Ethernet широко используется в локальных сетях.. Работа станций Ethernet заключается в следующем, когда станция решит, что ни одна другая рабочая станция не передает данные. При этом она продолжает «прослушивать» канал. Если обнаружена коллизия, то есть одновременная передача данных двумя рабочими станциями, то обе они прерывают процесс и перед повторной попыткой выжидают произвольный интервал времени. Если и далее возникают коллизии, то применяются другие алгоритмы. В одном из них время ожидания возрастает экспоненциально. Такая схема регулирования трафика называется *Множественный доступ с контролем несущей и обнаружением коллизий* (Carrier Sense Multiple Access with Collision Detection, CSMA/CD).

В каждом кадре Ethernet указан адрес, как отправители, так и получателя. Тем не менее, технология Ethernet широковещательная. Предполагается, что «честная» рабочая станция игнорирует кадры, адресованные не ей, но у сетевых адаптеров существует так называемый смешанный режим работы (promiscuous mode), когда все принимаемые станцией кадры, в том число и «чужие», передаются программному обеспечению верхнего уровня. Перенести станцию в этот режим могут как законные пользователи так и сетевые хакеры.

Обычно в сетях Ethernet используют топологии «шипа» или «звезда». Для соединения нескольких шип (или сетей) Применяются коммутаторы и мосты Ethernet. Они пересылают кадры, адресованные рабочим станциям и другой сети. Коммутаторы и мосты уменьшают зоны, в которых возникают коллизии, и дозволяют более эффективно использовать ресурсы сети для осмысленного трафика (того, что не приводит к коллизиям). Для соединения узлов Ethernet используют кабели различных типов.

***Gigabit Ethernet***

Примерно 100 компании создали объединение Gigabit Ethernet Alliance для разработки стандартов технологии Ethernet нового поколения. Gigabit Ethernet обеспечит передачу данных со скоростью 1 Гбит/с (гигабит в секунду) на расстояние от 300 м до 10 км. Очевидно, что сначала технологию Gigabit Ethernet внедряется на магистральных участках локальных или глобальных сетей. По всей вероятности, в крупных сетях строится трехуровневая модель, состоящая из Ethernet, Fast Ethernet и Gigabit Ethernet,

Под началом IEEE создай комитет 802.3z для формирования стандартов Gigabit Ethernet. Стандарт 802.3z требует реализации Gigabit Ethernet в физической среде одного из двух типов; неэкранированная витая пара (UTP) или оптоволоконный кабель. Сегодня, применяется только оптоволоконный кабель.

Текущие реализации Gigabil Ethernet используют полнодуплексный оптоволоконный канал, соединявший две станции. Другими словами, любой сегмент Ethernet состоит лишь из пары станции, каждая из которых имеет собственный канал передачи. Таким образом, исключена возможность возникновения коллизий. Стандартом Ethernet допускает искажение и повторную передачу не более одного кадра.

В стандарте 802.3z для увеличения допустимого расстояния между станциями описаны две технологии: увеличение нагрузки (carrier extension) и «выстреливание пакетов» (packet bursting). В первой короткие кадры искусственно удлиняются до 512 байт (напомним, что кадры Ethernet имеют длину от 64 байт). Это позволяет рабочим станциям обнаруживать коллизии, находясь на большом удалении друг от друга. Во втором варианте проблема решается за счет передачи кадров «очередями», сразу по несколько штук. Длина первого кадра «очереди» увеличивается до 512 байт, а последующие передаются без изменений.

***TokenRing***

Технология передачи маркера (token ring) содержит два основных стандарта. Первый — сеть TokenRing от IBM, для компании IBM она остается основной технологией локальных сетей. Второй — стандарт IEEE 802.5, разработанный Институтом инженеров и области электроники и электротехники (Institute Of Electrical and Electronics Engineers, IEEE). Он практически повторяет стандарт IBM TokenRing, однако, есть некоторые отличия. Основное в том, что спецификация IBM TokenRing требует топологию «звезда» и проводку па витой паре, а IEEE 802.5 не оговаривает этого.

Локальная сеть стандарта IЕЕЕ 802.5 основана па топологии «кольцо», а для определения очередности передачи информации используется короткое сообщение — маркер (token). Маркер циркулирует по кольцу, и только владеющая им рабочая станция может передавать данные. Рабочая станция, получившая маркер, но не желающая пересылать данные, просто отправляет его дальше. В сети с передачей маркера коллизии возникнуть не может. «Занятый» маркер (он имеет пометку «занято» и используется для переноса полезной информации) циркулирует до тех пор, пока не достигнет адресата. Затем он следует обратно к передающей рабочей станции, которая удаляет сообщение, снимает пометку и отправляет в кольцо уже «свободный» маркер.

Сеть IBM TokenRing использует точно такой же маркер, который в силу другой топологии проходит по иному пути. Вместо объединения в физическое кольцо, узлы подключаются в форме звезды к одному концентратору — MultiStation Access Unit (MSAU). В однонаправленном логическом кольце MSAU передает маркер от станции к станции по звездообразному маршруту. Каждый MSAU соединяет до 8 станции, а для удлинения логического кольца можно соединить до 33 MSAU.

Локальные сети с передачей маркера применяются там, где требуются гарантированная минимальная скорость передачи данных и некоторые возможности выявления и устранения неисправностей. Обычно такие сети обеспечивают скорости передачи данных от 4 до 16 Мбит/с.

***FDDI***

Оптоволоконная технологии FDDI (Fiber Distributed Data Interface) все чаще применяется в сетях — в основном для магистральных систем, соединяющих несколько локальных сетей. Благодаря скорости передачи данных 100 Мбит/с, FDDI также используется для высокоскоростной связи между большими и/или быстрыми компьютерами.

Стандарт FDDI, принятый ANSI (American National Standards Institute) и ISO. определяет физический уровень (модели ISO/OSI) с использованием оптоволоконного кабеля в локальной сети топологии «двойное кольцо». (Стандарты ANSI и ISO полностью совместимы между собой). Двойные кольца обеспечивают отказоустойчивость. Каждая станция FDDI подключена к двум кольцам, так называемым первичному и вторичному. При одиночном сбое возможно изолировать отказавшую станцию. Несколько сбоев иногда приводят к разбиению кольца на изолированные фрагменты, не способные к взаимодействию друг с другом. Также стандарт FDDI определяет протокол доступа к среде, в котором используется механизм передачи маркера. Каждая станция FDDI имеет уникальный 8-байтный адрес.

***Линии Т1/ТЗ***

Линии TI/T3 традиционно применяют для соединения двух значительно удаленных локальных сетей. Данные по TI проходят со скоростью 1,544 Мбит/с, а по ТЗ — до 44,763 Мбит/с. Линия Т1 состоит из 24 каналов по 64 кбит/с их можно использовать для разных целен. Например, по одним передавать голосовую информацию, а по другим какую-то еще, в том числе графику и видео. В настоящее время линии Т1 стали использоваться для подключения серверов Интернета к поставщикам услуг, особенно небольшими организациями, содержащими свои собственные Web-узлы.

***ISDN***

Широко применяемая операторами общественной телефонной связи (telephone carrier). Цифровая сеть интегрированных услуг (Integrated Services Digilal Nelwork, ISDN) — это семейство созданных CCITT протоколов, ориентированных на создание полностью цифровой всемирной сети передачи данных. Линия от подписчика до местной коммутационной станции, магистральные линии между коммутационными станциями и местная линия к адресату — цифровые, поэтому ISDN не требует ни одного аналого-цифрового преобразования. Кроме того, сеть ISDN обеспечивает большую полосу пропускания, чем обычная (аналоговая) телефонная сеть, и позволяет одновременно пересылать голосовые и другие данные (например, компьютерные, музыкальные или видео). Еще одно преимущество сети ISDN — высокая скорость установления соединения, она в 5 или б раз выше, чем для обычных телефонных линий.

В стандарт ISDN входит два различных протокола — BRI (Basic Rate Interface) и PRI (Primary Rate Interlace). При установке необходимо выбрать один из них. Стандарт PRI можно рассматривать как эквивалент линий T1. Но более распространен в сетях ISDN стандарт BRI, который содержит три раздельных канала:

• D-канал на 16 кбит/с (управляющий), для передачи управляющих и сигнальных данных ISDN;

• В-канал на 64 кбит/с (опорный), для передачи голосовой пли другой информации;

• второй В-канал для передачи голоса или других данных.

Принятый стандарт разуплотнения является расширением протокола РРР (Point-to-Point Protocol) и называется многоканальным РРР (multilink РРР). Однако многие производители ISDN-модемов предлагают собственные решения для разуплотнения канала, что вынуждает использовать одинаковые терминальные адаптеры (то есть ISDN-модемы) на обоих концах соединения.

***Frame relay***

Frame relay (ретрансляция кадров) — это протокол с коммутацией пакетов, применяемый в глобальных сетях. Он появился как часть ISDN, но был внедрен раньше формального принятия спецификаций ISDN.

 Технология Frame relay активно продвигается операторами телефонной связи и обычно используется в сочетании с линиями Т1, ТЗ или выделенными линиями на 56 кбит/с. Каждому подписчику назначается порт, состоящий из специального оборудования, обеспечивающего интерфейс между подписчиком и транспортом общего пользования. В процессе подписки между различными портами Frame relay устанавливаются постоянные виртуальные каналы (permanent virtual circuit, PVC), или постоянные логические соединения. Кроме того, каждому межпортовому каналу назначается фиксированная минимальная скорость передачи данных, гарантированная оператором связи. Порт может использовать скорости выше гарантированной, но в атом случае оператор связи лишь попытается передавать данные на более высокой скорости.

Протокол Frame relay ориентирован на создание логического соединения, каждый обслуживаемый вызов получает идентификатор канала данных (data link call identifier, DLCI), выделяемый динамически. Однако при появлении полупостоянный виртуальных каналов выделение DLCI происходит статически в момент их регистрации. Идентификатор DLCI включается во все последующие кадры данных. DLC1 имеет смысл только для локального порта, то есть для одного PVC, существующего между двумя станциями, возможны два идентификатора DLCI, причем отличные на разных портах.

Преимущество технологии Frame relay в том, что один порт может одновременно использовать несколько PVC для различных целен. Так, например, один PVC можно выделить для обслуживания исходящих телефонных звонков, что исключит загрузку всей емкости канала одним конкретным видом трафика, например входящим.

***SMDS***

SMDS (Switched Multimegabit Dala Service) — это высокоскоростная служба с коммутацией пакетов, способная переносить большие объемы данных со скоростями от 56 кбит/с до 34 Мбит/с. Она, так же как и Frame relay, широко внедряется операторами телефонии общего пользовании. В отличие от ATM и Frame relay, SMDS не устанавливает логического соединения, то есть она не основана па виртуальных каналах, один порт SMDS связывается с другим вызывая его предопределенный адрес, а маршрут информации заранее неизвестен. Общественные операторы связи предлагают дополнительные услуги, например возможность фильтрования соединений SMDS.

***ATM***

Технология сетей с асинхронным режимом передачи (Asynchronous Transler Mode, ATM) возникла при дополнении механизмами статистического уплотнения технологии синхронного режима передачи (Synchronous Transler Mode, STM). В ATM когда возникает необходимость передать данные, то они вместе с идентификатором соединения упаковываются в пустой пакет. Идентификатор соединения называют идентификатором виртуального канала (Virtual Circuit Identifier, VCI). В ATM добавлено статистическое уплотнение (динамическое перераспределение свободной емкости канала).

В ATM ограничений на количество соединений нет: считается, что не все установленные соединения одновременно передают данные, а если и так, то некоторые данные могут чуточку подождать в буфере до появления свободного пакета.

Технология ATM гарантирует поступление данных в порядке их отправки, но не гарантирует саму доставку. Сохранение порядка обеспечивается тем, что данные передаются по логическому каналу, то есть следуют по одному маршруту. Технология АТМ годится для самых равных приложении.

Технология ATM по своей сути ориентирована на логические соединения, но для совместимости со «старыми» приложениями, работающими без логических соединений, можно реализовать соответствующую службу поверх ATM. Такая служба сама выполняется как АТМ-приложение (то есть она не является составной частью ATM), имитируя модель серверов без логических соединении. Эти серверы используют содержащуюся в ячейках информацию о маршруте их следования и направляют данные к месту назначении.

***xDSL***

Технология DSL (Digital Subscriber Liny) позволяет увеличить скорость при существующих телефонных линиях, проведенных посредством медных кабелей, линия xDSL не является цифровой вследствие использования обычного медного провода, но на обоих концах установлены цифровые модемы. (Префикс хи — переменный, то есть в разных вариантах DSL он различный.) Преимущество xDSL по сравнению с ISDN в том, что даже при отключении электроэнергии в доме или офисе линия xDSL будет работать, так как это обычная телефонная линия

Медные провода способны переносить сигналы в гораздо более широком диапазоне частот, однако поскольку в медном проводе сигнал затухает достаточно быстро, то пропускная способность прямо зависит от длины линии. Так, например, провод 24-го калибра позволяет переносить данные при скорости 1,544 Мбит/с на расстояние 18 000 футов, а при скорости 51,840 Мбит/с — лишь на 1 000 футов.

Все описанные ниже технологии схожи в одном: по телефонному проводу передается высокочастотный сигнал, сгенерированный телефонной компанией, а технологии DSL лишь модулируют ого,

*DS1/T1/E1*

Технология DS1/T1/E1, изначально разработанная в Bell Labs, позволяет объединить двадцать каналов по 64 кбит/с в один высокоскоростной поток. Длина кадра в полученном канале составляет 193 бита, сюда входят и управляющие данные для уплотнения/разуплотнения. В настоящее время построено достаточно много таких линий, причем примерно через каждые 6 000 футов на них необходимо устанавливать повторители.

*DSL/ISDN*

Модемы DSL/ISDN способны пересылать данные со скоростью 160 кбит/с, а расстояние до 16000 футов. Как и в технологии ISDN, их пропускная способность разделена между двумя В-каналами, обеспечивающая ими скорость по 64 кбит/с, и одним D-каналом, обеспечивающим скорость 16 кбит/с.

*HDSL*

Высокоскоростная технология HDSL (High-Dala-Rale Diqital Subscriber Line) основана на современных разработках из области модуляции сигналов, что позволяет передавать данные быстрее и с меньшим числом повторителей. Эта технология применяется для соединения сотовых ретрансляционных станций и офисных АТС, для предоставления доступа в Интернет и создания частных сетей передачи данным. Две линии HDSL обеспечивают пропускную способность, эквивалентную каналу TI (1,544 Мбит/с), при рабочих расстояниях до 12 000 футов. Технология HDSL симметрична, то есть обеспечивает одинаковую скорость передачи данных в обоих направлениях.

*ADSL*

Асимметричная технология ADSL (Asymmetric Digital Subscriber Line) появилась позднее HDSL и предназначена для тех случаев, когда асимметричная загруженность канала выгодна. Она обеспечивает скорость приема (то есть из сети к потребителю) и диапазоне от 1,5 до 9 Мбит/с. Скорость передачи варьируется от 16 до 640 кбит/с. Большое преимущество ADSL по сравнению с ISDN в том, что питание в медный кабель идет от телефонной сети, — другими словами, пользоваться обычной телефонной связью можно даже при отключении электроэнергии. Технология ADSL стандартизирована институтами ANSI.

 В некоторых реализациях ADSL для модуляции линейного сигнала применяется амплитудно-фазовая модуляция с подавлением несущей (саr-rierless amplitude/phase, CAP). При использовании для передами данных САР-модуляции сигнал несущей частоты перед попаданием в линию связи подавляется. Эта модуляция не признана частью стандарта ADSL в ANSI.

В технологии ADSL есть другом способ модуляции — дискретная многочастотная модуляция (discrete multitone, DMT). При таком способе вся доступная полоса частот делится на 256 каналов, а поскольку высокочастотные сигналы более чувствительны к помехам, полученные каналы имеют разную пропускную способность. Модуляция DMT позволяет достичь более высоких скоростей, чем САР. Однако она принята ANSI в качестве стандарта для ADSL.

*RADSL*

Адаптивная технология RADSL (Rale-Adaptive Digital Subscriber Line) — это обычная ADSL, способная перед началом передачи протестировать качество и длину канала и подобрать подходящую скорость. Стандарты RADSL еще разрабатываются.

*VDSL*

Сверхскоростная технология VDSL (Very-High-Data-Rate DSL) является самой быстрой из семейства DSL. Она обеспечивает скорости от 13 до 52 Мбит/с при приеме и от 1,5 до 2,3 Мбит/с при передаче данных по обычным медным проводам на расстояниях от 1 000 до 4 000 футов. Однако это, вероятно, самая далекая от стандартизации технология семейства DSL.

*SDSL*

Одноканальная технология SDSL (Single-Line DSL) аналогична VDSL, но имеет два главных отличия:

• ограничена расстоянием 10 000 футов;

• использует лишь одну линию.

***Беспроводные сети***

Не ток давно IEEE завершил разработку спецификации IEEE 802.11, которая описывает беспроводные локальные сети и служит стандартом для их совместимости. В общих чертах беспроводная сеть состоит из устройств трех типов: клиент, сервер и точка доступа. Последняя, будучи приемно-передающим устройством, выполняет те же функции, что и ретранслятор сотовой связи, работая промежуточным звеном между клиентом и сервером, Клиент — это любое мобильное устройство, инициирующее запросы на передачу данных; в сервер — устройство, к которому клиент направляет свои запросы. Технология беспроводных сетей обеспечивает передачу данных на скорости от 1 до 2 Мбит/с при удаленности устройств на расстояния от 200 до 1 000 футов.

Спецификация IEEE 802.11 затрагивает как физический, так и канальный уровни модели ISO/OS1. В ней описай один стандарт для инфракрасных частот и два — дли радиочастот.

Стандарт IEEE 802.11 определяет только уровень MAC, что способствует лучшей совместимости и позволяет в будущем реализовать его и микросхемах, а затем расширить область применения и уменьшить стоимость локальных сетей стандарта IEEE 802.11. Этот уровень MAC использует множественный доступ с контролем несущей и предупреждением коллизий (CSMA/CA).

В самом стандарте описаны целых два алгоритма предупреждения коллизий. Первый использует точечную управляющую функцию (poinl coordination function, PCF) и в соответствии со спецификацией IEEE 802.11 не обязателен. Время работы РСF ограничено, так как периодически вместо нее используется распределенный интеллектуальный алгоритм доступа к среде, названный распределенная управляющая функция (distributed coordination function, DCF). Основной инструмент предупреждении коллизий для спецификации IEEE 802.11 — именно DCF. Спецификации IEEE 802.11 для уровня MAC требует, чтобы принимающая сторона подтверждала прием адресованных лично ей кадров.

***WLAN-сети, Wi-Fi***

В основе WLAN-технологий лежит принцип высокочастотной радиосвязи между узлами сети. В качестве узла сети может выступать как отдельный компьютер, ноутбук или PDA, так специальный устройство "точка доступа" или "Access Point" - обеспечивающее доступ к кабельному сегменту сети Ethernet, к Интернету или другому компьютеру.

Первый стандарт IEEE 802.11 для беспроводных локальных сетей был принят в 1997 году. Он подразумевал работу оборудования на частоте 2.4ГГц со скоростями 1 и 2 Мб/с. Стандарт разрабатывался в течение 7 лет и поэтому ко времени принятия уже не мог соответствовать выросшим потребностям.

Новый расширенный вариант стандарта, названный 802.11b (802.11 High Rate), был принят в 1999 году. С его принятием стала возможной работа беспроводных сетей на скоростях до 11Мб/с, что было сопоставимо по скорости с обычными сетями Ethernet. Такая скорость позволила существенно расширить область применения беспроводных сетей и поднять уровень задач, для которых стало возможным использование WLAN.

Торговая марка Wi-Fi гарантирует совместимость оборудования от разных производителей. Первоначально в ноутбуках использовались адаптеры стандарта 802.11b, поэтому логотип WiFi часто ассоциировался именно с этим стандартом.

В стандарте 802.11a используется OFDM схема модуляции сигнала - мультиплексирование с разделением по ортогональным частотам (Orthogonal Frequency Division Multiplexing). Основной поток данных разделяется на ряд параллельных подпотоков с относительно низкой скоростью передачи, и далее для их модуляции используется соответствующее число несущих. В стандарте определены три обязательные скорости передачи данных (6, 12 и 24 Мбит/с) и пять дополнительных (9, 18, 24, 48 и 54 Мбит/с). Также имеется возможность одновременного использования двух каналов, что обеспечивает увеличение скорости вдвое.

В стандарте 802.11b применяется метод широкополосной модуляции с прямым расширением спектра - DSSS (Direct Sequence Spread Spectrum). Весь рабочий диапазон делится на 14 каналов, разнесенных на 25 МГц для исключения взаимных помех. Данные передаются по одному из этих каналов без переключения на другие. Возможно одновременное использование всего 3 каналов. Скорость передачи данных может автоматически меняться в зависимости от уровня помех и расстояния между передатчиком и приемником.

Стандарт IEEE 802.11b обеспечивает максимальную теоретическую скорость передачи 11 Мбит/с, что сравнимо с обычной кабельной сетью 10 BaseT Ethernet. Однако, такая скорость возможна лишь при условии, что в данный момент только одно WLAN-устройство осуществляет передачу. При увеличении числа пользователей полоса пропускания делится на всех и скорость работы падает.

Стандарт 802.11g окончательно был ратифицирован в июне 2003г. Он является дальнейшей разработкой спецификации IEEE 802.11b и осуществляет передачу данных в том же частотном диапазоне. Основным преимуществом этого стандарта является увеличенная пропускная способность - скорость передачи данных составляет до 54 Мбит/с по сравнению с 11 Мбит/с у 802.11b. Как и IEEE 802.11b, новая спецификация предусматривает использование диапазона 2,4 ГГц, но для увеличения скорости применена таже схема модуляции сигнала - что и в 802.11a - ортогональное частотное мультиплексирование (OFDM).

Стандарт WLAN - 802.11n повышает скорость передачи данных практически вчетверо по сравнению с устройствами стандартов 802.11g, при условии использования в режиме 802.11n с другими устройствами 802.11n. Теоретически 802.11n способен обеспечить скорость передачи данных до 600 Мбит/с (стандарт IEEE 802.11ac до 1.3 Гбит/с), применяя передачу данных сразу по четырем антеннам. По одной антенне — до 150 Мбит/с.

Устройства 802.11n работают в диапазонах 2,4—2,5 или 5,0 ГГц.

Кроме того, устройства 802.11n могут работать в трёх режимах:

* наследуемом (Legacy), в котором обеспечивается поддержка устройств 802.11b/g и 802.11a;
* смешанном (Mixed), в котором поддерживаются устройства 802.11b/g, 802.11a и 802.11n;
* «чистом» режиме — 802.11n (именно в этом режиме и можно воспользоваться преимуществами повышенной скорости и увеличенной дальностью передачи данных, обеспечиваемыми стандартом 802.11n).

## 4. ПРОМЫШЛЕННЫЕ СЕТИ

**Промышленный Ethernet**

В настоящее время под Ethernet понимают семейство продуктов для локальных сетей, которые соответствуют стандарту IEEE 802.3. Промышленному применению стандарта долгое время мешал метод случайного доступа к сети, не гарантировавший доставку сообщения в короткое и заранее известное время. Однако это проблема была решена применением Недостатком промышленного Ethernet является относительно высокая цена: Ethernet модули ввода-вывода в среднем в 2 раза дороже аналогичных Modbus-устройств.

Внедрению Ethernet в промышленность способствовали следующие его качества: высокая скорость передачи (до 10 Гбит/с) и соответствие требованиям жесткого реального времени при высоком быстродействии; простота интеграции с Internet и Intranet, в том числе по протоколам прикладного уровня SNMP; простота интеграции с офисными сетями; наличие большого числа специалистов по обслуживанию Ethernet; по-настоящему открытые решения; возможность организации многомастерных сетей; неограниченные возможности по организации сетей самых разнообразных топологий; широкое применение в офисных сетях, что обеспечило экономическую эффективность технической поддержки стандарта со стороны международных организаций по стандартизации; появление недорогих коммутаторов, решивших проблему недетерминированности Ethernet.

Существуют 4 стандартные скорости передачи данных в сетях Ethernet по оптоволоконному кабелю и витой паре проводов: 10 Мбит/с, 100 Мбит/с, 1 Гбит/с и 10 Гбит/с.

*Ethernet имеет несколько модификаций, структура наименований которых имеет следующий вид: <скорость передачи>Base<дополнительные обозначения>. Скорость указывается цифрой в Мбит/с или в Гбит/с, в последнем случае к цифре добавляется буква G. Буквы после "Base" означают тип кабеля (T - "Twisted pair"- "витая пара", F - "Fiber optic" -"оптоволоконный", S - "Short wavelength optic" - "оптический коротковолновый", L - "Long wavelength" - "длинноволновый", C - "short Copper cable"). Символ "X" означает наличие блока кодирования на физическом уровне. В тексте стандарта IEEE 802.3-2005 [IEEE] приводится 44 варианта таких обозначений.* (10Base‑T означает спецификацию физического уровня для скорости 10 Мбит/с с методом доступа CSMA/CD и с использованием двух витых пар проводов; 100Base-FX - для скорости 100 Мбит/с, CSMA/CD с применением двух многомодовых оптических кабелей.)

Отличительными признаками промышленного Ethernet являются:

* отсутствие коллизий и детерминированность поведения благодаря применению коммутаторов;
* индустриальные климатические условия;
* устойчивость к вибрациям;
* отсутствие вентиляторов в оборудовании;
* повышенные требования к электромагнитной совместимости;
* компактность, крепление на ДИН-рейку;
* удобное подключение кабелей;
* диагностическая индикация на панели прибора;
* электропитание от источника напряжения в диапазоне от 10 до 30 В;
* возможность резервирования;
* разъемы и оборудование со степенью защиты до IP67;
* защита от электростатических зарядов, электромагнитных импульсов, от превышения напряжения питания;
* полнодуплексная передача.

Недостатком Ethernet является значительный уровень "накладных расходов" в протоколах TCP/IP, которые рассчитаны на пересылку данных больших объемов. Если же устройство пересылает всего 8 байт, что типично для АСУ ТП, то полезная информация в протоколе составляет всего 11%, поскольку каждый пакет TCP/IP содержится 68 байт служебной информации. Однако это часто можно скомпенсировать большой пропускной способностью сети.

**CAN (Controller Area Network)**

Стандарт CAN (Controller Area Network) ориентирована на последовательные каналы связи, выполненные из скрученных пар проводов (или оптических волокон), стандарт определяет протоколы физического уровня и субуровеней MAC и LLC. Все узлы сети равноправны и подключаются к общему каналу. Уровни сигналов протоколом не нормированы. В CAN использована кодировка типа NRZ (Non Return to Xero). Для распознавания сигнатур начала (SOF) и конца (EOF) кадра используется бит-стафинг.

Высокая надежность и дешевизна сделала сети CAN привлекательными для промышленности и науки. Сеть предназначена для сбора информации и управления в реальном масштабе времени, но может быть использована и для других целей. Канал can реализует принцип множественного доступа с детектированием столкновений (CSMA/CD - Carrier Sense Multiple Access with Collision Detection, аналогично Ethernet). Сеть может содержать только один сегмент. В соответствии со стандартом ISO 11898 сеть способна работать при обрыве одного из проводов, при закоротке одного из проводов на шину питания или на землю. Скорость работы канала программируется и может достигать 1 Мбит/с. Схема арбитража (выполняется побитно сравнение передаваемых кадров с использованием схемы проволочного “И”, при этом доминантным состоянием является логический “0”) позволяет сделать доступ к общему каналу существенно более эффективным, чем в случае Ethernet.

Протокол can имеет развитую систему диагностики ошибок. В результате вероятность не выявленной ошибки составляет менее 4.7 10-11. При выявлении ошибки кадр отбрасывается и его передача повторяется.

Число узлов, подключенных к каналу, протоколом не лимитируется. Практически такое ограничение налагается задержкой или предельной нагрузкой канала.

Любой модуль can может быть переведен в пассивный режим (“состояние сна”), при котором внутренняя активность прекращается, а драйверы отключаются от канала. Выход из этого режима возможен либо по внутренним причинам, либо вследствие сетевой активности. После “пробуждения” модуль ждет, пока стабилизируется его внутренний тактовый генератор, после чего производится синхронизация его работы с тактами канала (11 тактов). Благодаря синхронизации отдельные узлы не могут начать передачу асинхронно (со сдвигом на часть такта).

Время пребывания канала в пассивном состоянии не нормировано. Появление доминантного бита на шине, пребывающей в пассивном состоянии, воспринимается как начало очередного кадра. Предусматривается возможность установления масок в узле на отдельные двоичные разряды идентификатора, что позволяет игнорировать их значения. Маскирование делает возможным мультикастинг-адресацию.

**Profibus**

Слово PROFIBUS получено из сокращений PROcess FIeld BUS, что приблизительно переводится как "промышленная шина для технологических процессов". Стандарт Profibus был первоначально принят в Германии в 1987 году, затем, в 1996 году, он стал международным (EN 50170 и EN 50254).

Profibus имеет три модификации: Profibus DP, Profibus FMS и Profibus PA.

Profibus DP (Profibus for Decentralized Peripherals - "Profibus для децентрализованной периферии") в отличие от FMS и PA построен таким образом, чтобы обеспечить наиболее быстрый обмен данными с устройствами, подключенными к сети.

Profibus FMS (Profibus с FMS протоколом) применяется для обмена данными с контроллерами и компьютерами на регистровом уровне. Profibus FMS предоставляет большую гибкость при передаче больших объемов данных, но проигрывает протоколу DP в популярности вследствие своей сложности.

Profibus FMS и DP используют один и тот же физический уровень, основанный на интерфейсе RS-485 и могут работать в общей сети.

Profibus PA (Profibus for Process Automation - "для автоматизации технологических процессов") использует физический уровень на основе стандарта IEC 1158-2, который обеспечивает питание сетевых устройств через шину и не совместим с RS-485. Особенностью Profibus PA является возможность работы во взрывоопасной зоне.

В последние годы появился стандарт PROFInet, который основан на Industrial Ethernet и технологиях COM, DCOM. Он легко обеспечивает связь промышленной сети Profibus с офисной сетью Ethernet.

Profibus является многомастерной сетью (с несколькими ведущими устройствами). В качестве ведомых устройств выступают обычно устройства ввода-вывода, клапаны, измерительные преобразователи. Они не могут самостоятельно получить доступ к шине и только отвечают на запросы ведущего устройства.

На физическом уровне Profibus DP и FMS используют стандарт RS-485 при скорости передачи до 12 Мбит/с и с размерами сегментов сети до 32 устройств. Количество устройств можно увеличить с помощью повторителей интерфейса.

Для увеличения дальности передачи в Profibus предусмотрена возможность работы с оптоволоконным кабелем. При использовании стеклянного оптоволокна дальность связи может быть увеличена до 15 км. Оптоволоконные интерфейсы выполняются в виде сменных модулей для контроллеров.

Profibus DP он получил несравненно более широкое распространение, чем FMS и PA.

**Modbus**

Протокол Modbus и сеть Modbus являются самыми распространенными в мире.

Одним из преимуществ Modbus является отсутствие необходимости в специальных интерфейсных контроллерах, простота программной реализации и элегантность принципов функционирования. Высокая степень открытости протокола обеспечивается также полностью бесплатными текстами стандартов.

Modbus совместим с большим количеством оборудования, имеет высокую достоверность передачи данных, связанную с применением надежного метода контроля ошибок, позволяет унифицировать команды обмена.

Основным недостатком Modbus является сетевой обмен по типу "ведущий/ведомый", что не позволяет ведомым устройствам передавать данные по мере их появления и поэтому требует интенсивного опроса ведомых устройств ведущим.

Разновидностями Modbus являются протоколы Modbus Plus - многомастерный протокол с кольцевой передачей маркера и Modbus TCP, рассчитанный на использование в сетях Ethernet и интернет.

Протокол Modbus имеет два режима передачи: RTU (Remote Terminal Unit – «удаленное терминальное устройство») и ASCII. Стандарт предусматривает, что режим RTU в протоколе Modbus должен присутствовать обязательно, а режим ASCII является опционным.

Стандарт Modbus предусматривает применение физического интерфейса RS-485, RS-422 или RS-232. Наиболее распространенным для организации промышленной сети является 2-проводной интерфейс RS-485. Для соединений точка-точка может быть использован интерфейс RS-232 или RS-422.

Modbus-устройство обязательно должно поддерживать скорости обмена 9600 бит/с и 19200 бит/с, из них 19200 бит/с устанавливается "по умолчанию". Допускаются также скорости 1200, 2400, 4800,...,38400 бит/с, 65 кбит/с, 115 кбит/с,... .

Скорость передачи должна выдерживаться в передатчике с погрешностью не хуже 1%, а приемник должен принимать данные при отклонении скорости передачи до 2%.

Сегмент сети, не содержащий повторителей интерфейса, должен допускать подключение до 32 устройств, однако их количество может быть увеличено, если это допустимо исходя из нагрузочной способности передатчиков и входного сопротивления приемников, которые должны быть приведены в документации на интерфейсы. Указание этих параметров в документации является обязательным требованием стандарта.

Максимальная длина магистрального кабеля при скорости передачи 9600 бит/с и сечении жил более 0,13 кв. мм (AWG26) составляет 1 км. Отводы от магистрального кабеля не должны быть длиннее 20 м. При использовании многопортового пассивного разветвителя с N отводами длина каждого отвода не должна превышать значения 40 м/N.

Протокол Modbus предполагает, что только одно ведущее устройство (контроллер) и до 247 ведомых (модулей ввода-вывода) могут быть объединены в промышленную сеть. Обмен данными всегда инициируется ведущим. Ведомые устройства никогда не начинают передачу данных, пока не получат запрос от ведущего. Ведомые устройства также не могут обмениваться данными друг с другом. Поэтому в любой момент времени в сети Modbus может происходить только один акт обмена.

Адреса с 1 по 247 являются адресами Modbus устройств в сети, а с 248 по 255 зарезервированы. Ведущее устройство не должно иметь адреса и в сети не должно быть двух устройств с одинаковыми адресами.

В рамках семейства Modbus существует протокол Modbus TCP (или  Modbus TCP/IP), который используется для того, чтобы подключить устройства с протоколом Modbus к Ethernet или Internet сети. Он использует кадры Modbus RTU на 7-м (прикладном) уровне модели OSI, протоколы Ethernet на 1-м и 2-м уровне модели OSI и TCP/IP на 3-м и 4-м уровне, т. е. Ethernet TCP/IP используется для транспортировки модифицированного кадра Modbus RTU.

**Profinet**

PROFINET ориентирован на организацию системно-широкого обмена данными между всеми иерархическими уровнями управления предприятием. Он позволяет использовать существующие каналы связи и сетевые компоненты Ethernet, а также дополнять эти сети специализированными компонентами. PROFINET обеспечивает поддержку всех существующих стандартных механизмов обмена данными через Ethernet параллельно с обменом данными между системами автоматизации в реальном масштабе времени.

Для организации обмена данными между системами автоматизации в сети PROFINET могут использоваться электрические (витые пары), оптические и беспроводные каналы связи Ethernet. В зависимости от вида используемых каналов для построения сети может использоваться различный набор сетевых компонентов. Обеспечивается поддержка всех топологий, характерных для сети Industrial Ethernet: линейных, кольцевых, древовидных.

В настоящее время наиболее ярко прослеживаются два направления использования сетей PROFINET: построение систем распределенного ввода-вывода (PROFINET IO) и построение модульных систем управления с распределенным интеллектом – PROFINET CBA (Component Based Automation).

В системах PROFINET IO приборы полевого уровня подключаются непосредственно к сети Industrial Ethernet и обслуживаются PROFINET контроллером ввода-вывода. Скоростной обмен данными носит циклический характер и выполняется на скорости 100 Мбит/с.

 В зависимости от состава используемых компонентов в такой сети обеспечивается поддержка обмена данными в реальном масштабе времени (Real Time – RT) и использование тактовой синхронизации (Isochronous RT – IRT).

Поддерживается возможность интеграции существующих сетей PROFIBUS DP в системы PROFINET IO. При этом ведущее устройство подключается к сети PROFINET и поддерживает связь с ведомыми устройствами PROFIBUS DP через PROFINET Proxy.

# ПРОТОКОЛЫ ПЕРЕДАЧИ ДАННЫХ

## КАНАЛЬНЫЙ УРОВЕНЬ ПРОМЫШЛЕННЫХ ПРОТОКОЛОВ

### Profibus DP

Канальный уровень модели OSI в Profibus называется FDL - уровнем (Fieldbus Data Link - "промышленный канал связи"). Объект MAC (Medium Access Control - "управление доступом к каналу") на канальном уровне определяет процедуру передачи данных устройствами, включая управление правами на передачу данных через сеть. Протокол канального уровня обеспечивает выполнение следующих важных требований:

* в процессе коммуникации между ведущими устройствами необходимо обеспечить выполнение каждым из них своей задачи в течение заранее определенного интервала времени;
* взаимодействие ведущих устройств (контроллеров) с ведомыми должно происходить максимально быстро.

В обычном режиме центральный контроллер (ведущий) периодически считывает информацию, поступающую на входы ведомых устройств и записывает информацию для их выходов. Дополнительно к этому периодическому обмену данными DP обеспечивает мощные средства для диагностики и инсталляции системы, а также для обеспечения устойчивости к внешним дестабилизирующим факторам.

Средства диагностики DP обеспечивают быстрое обнаружение места появления ошибки и пересылку соответствующего сообщения ведущему устройству. Диагностические сообщения делятся на три уровня:

* уровень устройства (касающиеся общих признаков работоспособности устройства, таких как перегрев, уход напряжения за допустимые границы и др.);
* уровень модуля ввода-вывода, входящего в состав устройства модульной конструкции;
* уровень канала модуля (например, "к. з. входа 8").

К одной сети могут быть подсоединены до 128 устройств (но не более 32-х в одном сегменте). Спецификация для конфигурирования системы включает количество узлов сети, распределение адресов устройств, формат диагностических сообщений, параметры шины.

В сети могут использоваться устройства трех типов:

* DP мастер класса 1 (DPM1) - центральный контроллер, который циклически обменивается информацией с ведомыми устройствами с заранее определенным периодом;
* DP мастер класса 2 (DPM2) - устройство, предназначенное для конфигурирования системы, наладки, обслуживания или диагностики;
* ведомое устройство - устройство, которое выполняет сбор информации или выдачу ее исполнительным устройствам.

Эти устройства могут отсылать или принимать не более 256 байт информации за один цикл обмена.

Коммуникационный профиль DP позволяет сконфигурировать как одномастерную, так и многомастерную сеть. В одномастерной сети ведущее устройство (мастер) может посылать запросы и получать ответы только от ведомых устройств.

В многомастерной сети имеется несколько ведущих устройств, которые имеют свои одномастерные подсети и в пределах подсети являются устройствами класса DPM1. Ведущие устройства в многомастерной сети могут быть также устройствами класса DPM2. Входные и выходные данные подчиненных устройств могут быть прочитаны любым мастером сети. Однако записывать данные в устройства может только один мастер, который при конфигурировании системы был обозначен как DPM1.

Ведущий контроллер (DPM1) может находиться в одном из трех состояний: Stop -  когда не происходит обмена данными; Clear  - когда DPM1 может считывать данные, но не может записывать их и выходы всех устройств переводятся в безопасные состояния; Operate - обычное рабочее состояние.

Если в системе появляется сообщение об ошибке, то DPM1 устанавливает выходы всех устройств вывода в безопасное состояние, а сам переходит в состояние "Clear". *Безопасным* считается такое состояние, при котором исполнительные устройства находятся в безопасном (для человека или системы) состоянии. Такое состояние самоконтроля системы может быть установлено или нет при ее конфигурировании. При отключенном состоянии самоконтроля система продолжает работать несмотря на появление ошибок.

При конфигурировании системы пользователь назначает каждому ведущему свои ведомые устройства и очередность их опроса, а также указывает устройства, которые не надо опрашивать периодически.

Передача данных между мастером DPM1 и ведомыми делится на три фазы: параметризация, конфигурирование и передача данных. В фазе параметризации и конфигурирования проверяется, соответствует ли конфигурация и параметры ведомого устройства запланированным в DPM1 установкам. Проверяется тип устройства, формат и длина передаваемых сообщений, количество входов или выходов.

Profibus DP имеет режим синхронизации вывода. Для этого посылается широковещательная управляющая команда синхронизации, при получении которой происходит одновременная смена состояний выходов всех устройств вывода. Имеется также команда "замораживание", при поступлении которой входы всех устройств ввода сохраняют свое текущее состояние и перестают реагировать на изменение поступающих на входы сигналов, пока не поступит команда "размораживание". Эти команды используются для синхронизации ввода. Команды синхронизации могут посылаться всем устройствам сети, группе или одному устройству.

Для обнаружения ошибок в передающих устройствах предусмотрен механизм временного мониторинга (наблюдения), который действует как в ведомых, так и ведущих устройствах. Интервал мониторинга устанавливается при конфигурировании системы.

Ведомый также выполняет контроль ведущего устройства или линии передачи. Для этого используется сторожевой таймер. Если от ведущего не приходят данные в течение периода сторожевого таймера, ведомый автоматически переводит свои выходы в безопасные состояния. Для большей степени защиты выходов в многомастерной системе только один (уполномоченный) мастер имеет прямой доступ к изменению состояний выходов устройства. Все другие мастера могут считывать только "изображения" сигналов на входах и выходах устройства.

Передача сообщений

Profibus использует два типа сервисов для передачи сообщений: SRD (Send and Receive Data with acknowledge - "отправка и прием данных с уведомлением") и SND (Send Data with No acknowledge - "отправка данных без уведомления").

Сервис SRD позволяет отправить и получить данные в одном цикле обмена. Этот способ обмена наиболее распространен в Profibus и очень удобен при работе с устройствами ввода-вывода, поскольку в одном цикле можно и отправить, и получить данные.

Сервис SND используется, когда надо отправить данные одновременно группе ведомых устройств (многоабонентский режим) или всем ведомым устройствам (широковещательный режим). При этом ведомые устройства не отправляют свои уведомления мастеру.

Сообщение в Profibus называется телеграммой. Телеграмма может содержать до 256 байт, из них 244 байта данных, плюс 11 служебных байт (заголовок телеграммы). Все телеграммы имеют заголовки одинаковой длины, за исключением телеграммы с названием Data\_Exchange. Заметим, что 11 байт служебной информации делают Profibus очень неэффективным при передаче коротких сообщений. Однако при больших объемах данных такой формат телеграммы достаточно эффективен.



Поля телеграммы на [рис. 2.24](http://bookasutp.ru/Chapter2_7.aspx#%D1%80%D0%B8%D1%81.%202.24) имеют следующее содержание:

* SD - стартовый разделитель. Используется для указания начала телеграммы и ее формата. Имеется четыре типа разделителей для телеграмм запроса и ответа и один тип для короткого уведомления. Короткое уведомление имеет поле SD, но не в начале телеграммы;
* LE - длина передаваемых данных (DA+SA+FC+DSAP+SSAP+DU);
* LEr - повторение поля LE с целью его резервирования;
* DA - адрес устройства-получателя телеграммы;
* SA - адрес отправителя;
* FC - код типа телеграммы (запрос, уведомление, ответ, диагностические данные, тип устройства - мастер или ведомый, приоритет, уведомление);
* DSAP - устройство-получатель использует это поле чтобы определить, какой тип сервиса нужно выполнить;
* SSAP - COM порт отправителя;
* DU - данные длиной от 1 до 244 байт;
* FCS - контрольная сумма телеграммы (сумма значений полей DA+SA+ FC+DU, по модулю 255);
* ED - признак конца.

С целью повышения надежности в Profibus предусмотрено резервирование, выполненное следующим образом:

* ведомые устройства содержат два различных Profibus-интерфейса, основной и резервный. Они могут быть либо в одном устройстве, либо в двух одинаковых устройствах (основном и резервном);
* устройства снабжаются двумя независимыми стеками протоколов со специальным расширением для резервирования;
* процесс резервирования стеков протоколов осуществляется путем запуска специального программного объекта резервирования RedCom.

В нормальном режиме коммуникация выполняется только через основное устройство, которое посылает диагностическую информацию резервному устройству.

**CAN**

Сеть CAN ориентирована на последовательные каналы связи, стандарт определяет протоколы физического уровня и субуровеней MAC и LLC. Все узлы сети равноправны и подключаются к общему каналу. Уровни сигналов протоколом не нормированы. В CAN использована кодировка типа NRZ (Non Return to Xero). Для распознавания сигнатур начала (SOF) и конца (EOF) кадра используется бит-стафинг.

Когда канал свободен, любой из подключенных узлов, может начать передачу. Кадры могут иметь переменную, но конечную длину. Формат информационного кадра сети CAN, содержащего семь полей, показан на рис. 4.1.4.1.



Рис. 4.1.4.1 Стандартный информационный кадр 1 2.0a CAN

Кадр начинается с *доминантного бита* начала кадра (логический нуль, SOF - start of frame). Далее следует *поле арбитража* (идентификатор кадра), содержащее 11 бит (эти разряды носят имена id-28, ..., id-18) и завершающееся битом RTR (remote transmission request) удаленного запроса передачи. В *информационном кадре* RTR=0, для кадра запроса RTR=1. Семь наиболее значимых бит id-28 - id-22 не должны быть никогда все одновременно равными 1. Первым передается бит id28. Доминантные биты r0 и r1 (=0) зарезервированы для будущего использования. Поле *DLC* (Data Length Code) биты поля имеют имена dcl3 - dcl0) несет в себе код длины поля данных в байтах. *Поле данных*, размещенное вслед за ним, может иметь переменную длину или вообще отсутствовать. CRC - циклическая контрольная сумма. В качестве образующего полинома при вычислении CRC используется x15 + x14 + x10 + x8 + x7 + x4 + x3 + 1. Формально, следующий за контрольной суммой бит-разграничитель (=1) принадлежит полю CRC. Поле отклика (*ACK*) содержит два бита, первый из которых первоначально имеет уровень 1, а узлы получатели меняют его значение на доминантное (логический 0). Бит используется для сообщения о корректности контрольной суммы. Второй бит поля всегда имеет уровень логической 1. Завершающее поле *EOF* (end of frame) содержит семь единичных бит. За этим полем следует поле-заполнитель (*INT*) из трех единичных бит, после него может следовать очередной кадр. Формат расширенного информационного кадра сети can показан на рис. 4.1.4.2.



Рис. 4.1.4.2. Расширенный информационный кадр 2.0b CAN

Однобитовое субполе *SRR* (substitute remote request) включено в поле арбитража (идентификатора кадра) и всегда содержит код 1, что гарантирует преимущество стандартного информационного кадра (2.0a) случае его соревнования с расширенным кадром (2.0b) (при равных 11 битах идентификатора). Субполе *IDE* (identifier extension) служит для идентификации расширенного формата и для этого типа кадра всегда имеет уровень логической 1. Вслед за IDE следует 18-битовое поле (биты имеют имена id-17, ..., id-0. Контроллеры 2.0b полностью совместимы с кадрами 2.0a и могут посылать и принимать пакеты обоих типов. Идентификаторы в пределах одной сети должны быть уникальными. Следует иметь в виду, что 18-битное поле расширения идентификатора можно при определенных условиях использовать и для передачи информации. Идентификатор не является адресом места назначения, а определяет назначение передаваемых данных (адресация по содержанию). По этой причине пакет может быть принят отдельным узлом, группой узлов, всеми узлами сети или не воспринят вообще. Предельное число различных идентификаторов для версии 2.0a составляет 2032, а для 2.0b превышает 500 миллионов.

Послав кадр-запрос другому узлу, отправитель может затребовать присылку определенной информации. Удаленный узел должен откликнуться информационным кадром с тем же идентификатором, что и запрос.

Если несколько узлов начнут передачу одновременно, право передать кадр будет предоставлено узлу с более высоким приоритетом, который задается идентификатором кадра. Механизм арбитража гарантирует, что ни информация, ни время не будут потеряны. Если одновременно начнется передача запроса и информационного кадра с равными идентификаторами, предпочтение будет дано информационному пакету. В процессе арбитража каждый передатчик сравнивает уровень передаваемого сигнала с реальным уровнем на шине. Если эти уровни идентичны, он может продолжить передачу, в противном случае передача прерывается и шина остается в распоряжении более приоритетного кадра.

Протокол can имеет развитую систему диагностики ошибок.

Число узлов, подключенных к каналу, протоколом не лимитируется. Практически такое ограничение налагается задержкой или предельной нагрузкой канала.

Информационные кадры и удаленные запросы могут использовать как стандартные, так и расширенные форматы кадров (2.0a и 2.0b).

**Кадр удаленного запроса** может иметь стандартный и расширенный форматы. В обоих случаях он содержит 6 полей: SOF, поле арбитража, поле управления, CRC, поля ACK и EOF. Для этого типа кадров бит RTR=1, а поле данных отсутствует вне зависимости от того, какой код содержится в субполе длины.

**Кадр сообщения об ошибке** имеет только два поля - суперпозиция флагов ошибки и разграничитель ошибки. Флаги ошибки бывают активными и пассивными. Активный флаг состоит из шести нулевых бит, а пассивный - из шести единиц. Суперпозиция флагов может содержать от 6 до 12 бит. Разграничитель ошибки состоит из восьми единичных бит.

**Кадр перегрузки** включает в себя два поля - перпозиция флагов перегрузки и разграничитель перегрузки (8 бит =1). В поле *флаг ерегрузки* записывается 6 бит, равных нулю (как и в поле *флаг активной ошибки*). Кадры ошибки или перегрузки не требуют межкадровых разделителей. Существует ряд условий перегрузки, каждое из которых вызывает посылку такого кадра:

* внутренние обстоятельства приемника, которые требуют задержки передачи следующего кадра данных или запроса.
* Детектирование доминантного бита в начале поля *int*.
* Обнаружение узлом доминантного восьмого бита в поле *разграничителя ошибки* или *разграничителя перегрузки*.

Время пребывания канала в пассивном состоянии не нормировано. Появление доминантного бита на шине, пребывающей в пассивном состоянии, воспринимается как начало очередного кадра. Предусматривается возможность установления масок в узле на отдельные двоичные разряды идентификатора, что позволяет игнорировать их значения. Маскирование делает возможным мультикастинг-адресацию.

Поля *SOF, идентификатор, управляющее поле, данные* и *CRC* кодируются таким образом, что при появлении пяти идентичных бит подряд, в поток вставляется бит противоположного уровня. Так 0000000 преобразуется в 00000100, а 1111110 в 11111010. Это правило не распространяется на CRC-разделители, поля ACK и EOF, а также на кадры сообщения об ошибке или переполнении.

Номинальное время, выделенное для передачи одного бита, включает в себя четыре временные области: sync\_seg, prop\_seg, phase\_seg1, phase\_seg2 (рис.3.4.4.3).



Рис. 4.1.4.3 Временные зоны периода передачи одного бита

Первая временная область (SYNC\_SEG) служит для синхронизации работы различных узлов сети. Область PROP\_SEG предназначена для компенсации временных задержек в сети и равна сумме времени распространения сигнала по каналу и задержки во входных компараторах. PHASE\_SEG1 и PHASE\_SEG2 служат для компенсации фазовых ошибок и могут увеличиваться или уменьшаться после синхронизации. T0 - минимальный квант времени, используемый для формирования временной шкалы в пределах периода передачи одного бита (длительность внутреннего такта может быть значительно короче). Момент стробирования определяет момент времени, когда проверяется состояние канала. Этот момент должен быть синхронным для всех узлов сети. Длительность этих временных областей может задаваться программно. Чем длиннее канал, тем меньшую скорость передачи информации он может обеспечить.

### Modbus

Протокол Modbus имеет два режима передачи: RTU (Remote Terminal Unit – «удаленное терминальное устройство») и ASCII. Стандарт предусматривает, что режим RTU в протоколе Modbus должен присутствовать обязательно, а режим ASCII является опционным. Пользователь может выбирать любой из них, но все модули, включенные в сеть Modbus, должны иметь один и тот же режим передачи.

В стандарте Modbus имеются *обязательные* требования, *рекомендуемые* и *опционные* (необязательные). Существует три степени соответствия стандарту: «полностью соответствует» - когда протокол соответствует всем обязательным и всем рекомендуемым требованиям, «условно соответствует» - когда протокол соответствует только обязательным требованиям и не соответствует рекомендуемым, и «не соответствует».

Протокол Modbus предполагает, что только одно ведущее устройство (контроллер) и до 247 ведомых (модулей ввода-вывода) могут быть объединены в промышленную сеть. Обмен данными всегда инициируется ведущим. Ведомые устройства никогда не начинают передачу данных, пока не получат запрос от ведущего. Ведомые устройства также не могут обмениваться данными друг с другом. Поэтому в любой момент времени в сети Modbus может происходить только один акт обмена.

Адреса с 1 по 247 являются адресами Modbus устройств в сети, а с 248 по 255 зарезервированы. Ведущее устройство не должно иметь адреса и в сети не должно быть двух устройств с одинаковыми адресами.

В протоколе Modbus RTU сообщение начинает восприниматься как новое после паузы (тишины) на шине длительностью не менее 3,5 символов (14 бит), т. е. величина паузы в секундах зависит от скорости передачи.

|  |
| --- |
|  |
| Рис. 2.25. Формат кадра протокола Modbus RTU; PDU - "Protocol Data Unit" - "элемент данных протокола";  ADU - "Application Data Unit" - "элемент данных приложения" |

Формат кадра показан на рис.2.25. Поле адреса всегда содержит только адрес ведомого устройства, даже в ответах на команду, посланную ведущим. Благодаря этому ведущее устройство знает, от какого модуля пришел ответ.

*Поле «Код функции»* говорит модулю о том, какое действие нужно выполнить.

*Поле «Данные»* может содержать произвольное количество байт. В нем может содержаться информация о параметрах, используемых в запросах контроллера или ответах модуля.

*Поле «Контрольная сумма»* содержит контрольную сумму CRC длиной 2 байта.

В режиме RTU данные передаются младшими разрядами вперед.

По умолчанию в RTU режиме бит паритета устанавливают равным 1, если количество двоичных единиц в байте нечетное, и равным 0, если оно четное. Такой паритет называют четным (even parity) и метод контроля называют контролем четности.



|  |
| --- |
|  |

Рис. 2.26. Последовательность битов в режиме RTU; МЗР – младший значащий разряд. При отсутствии бита паритета на его место записывается второй стоп-бит

При четном количестве двоичных единиц в байте бит паритета может быть равен 1. В этом случае говорят, что паритет является нечетным (odd parity).

Контроль четности может отсутствовать вообще. В этом случае вместо бита паритета должен использоваться второй стоповый бит. Для обеспечения максимальной совместимости с другими продуктами рекомендуется использовать возможность замены бита паритета на второй стоповый бит.

Ведомые устройства могут воспринимать любой из вариантов: четный, нечетный паритет или его отсутствие.

Сообщения Modbus RTU передаются в виде кадров, для каждого из которых известно начало и конец. Признаком начала кадра является пауза (тишина) продолжительностью не менее 3,5 шестнадцатеричных символов (14 бит). Кадр должен передаваться непрерывно. Если при передаче кадра обнаруживается пауза продолжительностью более 1,5 шестнадцатеричных символа (6 бит), то считается, что кадр содержит ошибку и должен быть отклонен принимающим модулем. Эти величины пауз должны строго соблюдаться при скоростях ниже 19200 бит/с, однако при более высоких скоростях рекомендуется использовать фиксированные значения паузы, 1,75 мс и 750 мкс соответственно.

В режиме RTU имеется два уровня контроля ошибок в сообщении:

* контроль паритета для каждого байта (опционно);
* контроль кадра в целом с помощью CRC метода.

CRC метод используется независимо от проверки паритета. Значение CRC устанавливается в ведущем устройстве перед передачей. При приеме сообщения вычисляется CRC для всего сообщения и сравнивается с его значением, указанным в поле CRC кадра. Если оба значения совпадают, считается, что сообщение не содержит  ошибки.

Стартовые, стоповые биты и бит паритета в вычислении CRC не участвуют.

## СЕТЕВОЙ УРОВЕНЬ

### IP

***IP-адресация***

В протоколе IP версии 4 (IPv4) сетевые адреса имеют длину 32 бита. Распределением IP-адресов занимается организация IANA (Internet Assigned Numbers Authority). Все множество IP-адресов разделено на несколько категории, называемых классами адресов (табл. 3-1).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Старшие биты IP-адреса** | **Маска подсети** | **Число возможных сетей** | **Число возможных хостов** |
| Класс A | 0 | 255.0.0.0 | 126 | 16777214 |
| Класс B | 10 | 255.255.0.0 | 16384 | 65534 |
| Класс C | 110 | 255.255.255.0 | 2097152 | 254 |
| Класс D | 1110 | Нет  | Нет | Нет |

Классы IP-адресов.

Обычно Ip-адреса представлены в десятичном формате с точками (dotted decimal) разделяющими байты (со значениями от 0 до 255), например: 255.25.25.25. С каждым IP-адресом связана 32-разрядная маска подсети, которая при побитном умножении на IP-адрес делит последний на две части. Первая — уникальный идентификатор сети (адрес сети), вторая — уникальный идентификатор хоста в пределах данной сети (адрес хоста).

Организация, получающая адреса класса В, не обязана размещать все 16 000 компьютеров в одном месте. Можно составить подсети из пары сотен компьютеров в каждой. Для этого надо просто переопределить маску подсети, увеличив в ней количество бит, равных 1. Этот процесс называют делением па подсети (subnetting); он позволяет сократить количество адресов, доступных для компьютеров и маршрутизаторов за счет увеличения максимального числа возможных подсетей.

Некоторые IP-адреса имеют специальное назначение и не могут быть выданы компьютерам или маршрутизаторам, например адрес, все биты которого равны 0 или 1, а также некоторые другие.

Класс D IP-адресов зарезервирован дли группового вещания (multicast). К нему относятся IP-адреса от 224.0.0.0 до 239.255.255.255. Такие IP-адреса представляют логические группы узлов, причем сами узлы (несколько компьютеров и/или маршрутизаторов) могут территориально находиться далеко друг от друга. Так же как в обычной адресации, некоторые групповые IP-адреса зарезервированы и имеют специальное назначение. Например, 224.0.0.1 обозначает все системы дайной подсети, а 224.0.0.2 — все маршрутизаторы данной подсети. Подробнее об атом — в RFC 1918, 1519, 1518 и 1466.

IР-датаграммы

IР-датаграмма состоит из IP-заголовка и данных, которые часто называют *полезной нагрузкой* (payload). Длина IР-заголовка не фиксирована (рис. 3-1).

|  |  |  |  |
| --- | --- | --- | --- |
| Версия**(4 бита)** | Длина заголовка **(4 бита)**  | Тип обслуживания**(8 бит)** | Длина**(16 бит)** |
| Идентификатор | Флаги | Смещение фрагмента |
| Время жизни | Идентификатор протокола | Контрольная сумма |
| IP-адрес отправители |
| IP-адрес получателя |
| Параметры | Выравнивание |

*Рис.* 3.1. *IP-заголовок.*

Поле *Версия* (Version) состоит из четырех бит и обычно равно 4, обозначая четвертую версию протокола IP.

Поле *Длина заголовка* (Internet Header Length, IHL) состоит из 4 бит и содержит длину IP-заголовка в 32-разрядных слонах. Минимальная длина IP-заголовка составляет 20 байт.

Поле *Тип Обслуживания* (Type Of Service, TOS) состоит из 8 бит, в нем указывают необходимость специальной обработки пакета, например минимизации задержки, повышения скорости и т. д.

Поле *Длина* (Length) состоит из 16 бит и характеризует общую длину (в байтах) IP-пакета. Из зтого следует, что IP-пакеты не могут быть длиннее 65 535 байт, а поскольку IP-заголовок занимает как минимум 20 бант, то полезная нагрузка составит не более 65 515 байт.

Сами IP-датаграммы вкладываются в кадры протоколов нижнего уроня, например Ethernet или TokenRing, и могут пересылаться череп разные сети, в которых реализации протоколов нижних уровней зачастую отличаются.

Один компьютер способен передавать много IP-датаграмм и все они могут быть фрагментированы. Поэтому одного IP-адреса отправителя недостаточно для идентификации пакетов. Здесь помог бы номер порта, но он существует только в протоколах верхних уровней, например в UDP или TCP. Именно в такой ситуации используется поле *Идентификатор* (Identification). Оно имеет одинаковое значение во всех фрагментах, составляющих одну IP-датаграмму, и определяется хостом-отправителем.

Поле *Смещение фрагмента* (Fragment Offset) состоит из 13 бит и задает место фрагмента при сборки исходного потока данных. Дли первого фрагмента значение смещения равно 0. Не забывайте, что разные фрагменты передаются как отдельные IP-пакеты и, возможно, по разным маршрутам. Следовательно, окончательная сборка выполняется только адресатом, а не промежуточным маршрутизатором.

Поле *Флаги* (Flags) состоит из 3 бит. Старший бит всегда имеет значение 0, Следующий бит называется «Don't Fragment» (He фрагментировать) и, установленный в 1, говорит о недопустимости фрагментации данной датаграммы. Последний бит— «More Fraqmenls» (Еще фрагменты) — равен 1 во всех фрагментах одной датаграммы, кроме последнего.

Поле *Время жизни* (Time To Live, TTL) используется несколько иначе, чем предполагалось. В настоящее время это поле используют в качестве счетчика количества узлов (хостов, маршрутизаторов и т. п.), пройденных пакетом, то есть как счетчик переходов (hop count). Каждый узел уменьшает его на единицу, а при достижении 0 пакет уничтожается. По умолчанию его значение равно 32, однако отправитель (то есть его IP) может присвоить ему любое значение.

Поле *Идентификатор* протокола (Protocol ldenеtifier) состоит из 8 бит и показывает, какой протокол верхнего уровня вложен в поле данных IP-пакета. Это поле используется в IP для выбора одного из протоколов верхнего уровня, которому следует передать принятый пакет, например UDP или TCP.

Поле *Контрольная сумма* (Checksum) применяется для выявления ошибок передачи пакета. Здесь контрольная сумма — это двоичное дополнение 16-разрядной суммы по содержимому заголовка.

Поле *Параметры* (Options) имеет переменную длину и может вообще отсутствовать (длина 0 тоже допустима). Возможны два формата этого поля:

• один байт — тип параметра;

• несколько байт — тип параметра, длина и знамение.

Тип параметра состоит из трех подполей.

• Флаг COPIED (1 бит). Значения 1 или 0 свидетельствуют о том, необходимо или нет скопировать параметр во все фрагменты датаграммы.

• Класс параметра (2 бита). Значение 0 — контроль, 2 — отладка. Значения 1 и 3 зарезервированы.

• Номер параметра (5 бит).

Поле *Выравнивание* (Padding) — это лишь дополнительно нанятое место, добавленное для того, чтобы IP-заголовок заканчивался на 32-ом байте. Например, если поле Параметры заканчивается на 30-ом байте, то поле Выравнивание состоит из 2 байт, а если поле Параметры заканчивается на 50-ом байте, то Выравнивание — из 14 байт.

### ICMP

Протокол ICMP (Internet Control Message Protocol) используется хостами и маршрутизаторами для обмена управлявшей информацией, например, об ошибках или о процессе начальной загрузки. На рис. показан формат ICMP-сообщений и способ их вложения в IP-пакеты.

|  |
| --- |
| IP-заголовок. |
| ICMP-тип | ICMP-код | Контрольная сумма ICMP |
| Данные ICMP-сообщения |

Рис. Формат ICMP-сообщений.

Все биты, непосредственно относящиеся к ICMP-сообщению, являются полезной нагрузкой IP-пакета. В IP-заголовке поле *Идентификатор протокола* устанавливается равным 1. ICMP хотя и рассматривается как самостоятельный прикладной протокол, появляется неотъемлемой частью любой реализации протокола IP, поскольку управляет его работой.

Некоторые ICMP-сообщения генерируются только маршрутизаторами, а остальные — и маршрутизаторами, и хост-компьютерами. Никогда ICMP -сообщения не отправляются в ответ на другие ICMP-сообщения. Так предотвращается рекурсия ICMP-сообщений. Также ICMP-сообщения не генерируются в следующих случаях:

• если принято ICMP-сообщение об ошибке;

• если принята дейтаграмма с адресом локальной заглушки (loopback) либо групповым адресом;

• если принят не начальный IP-фрагмент.

Существует два типа ICMP-сообщений; ICMP -сообщения об ошибках и ICMP-запросы. Подробности о ICMP-сообщениях и полное описание их формата — в RFC 1256, 1122, 792 и др. Далее в этом разделе описаны только наиболее важны ICMP -сообщения.

Обычно маршрутизаторы посылают ICMP-сообщения для отчета об ошибках. Если маршрутизатор не обрабатывает датаграмму из-за недостатка ресурсов (например, памяти), то посылает ее отправителю ICMP-сообщение «Подавление источника» (source quench). Если маршрутизатор получает пакет данных для неизвестного адресата, то посылает ICMP-сообщение «Адресат недоступен» (destination unreachable). Чтобы рекомендовать хосту другой, более предпочтительный маршрутизатор для передачи данных к некоторому адресату, маршрутизатор посылает ICMP-сообщение, именуемое «Перенаправление» (redirect). В ICMP-сообщения об ошибках включается заголовок IP-датаграммы, вызвавшей ошибку. Это позволяет ее отправителю выяснить, с какими адресатами возникла ошибка, так как иногда одновременно проводятся несколько разных сеансов связи. Хосты обычно посылают ICMP-сообщения для проверки качества связи и оценки времени отзыва. Для этого используется ICMP-сообщение «Эхо» (echo). Хост, получивший такое сообщение, должен ответить ICMP-сообщением «Эхо-ответом» (echo reply).

### IGMP

Узел с обычным (не групповым) IP-адресом может самостоятельно входить и выходить из состава широковещательной группы (multicasl group). Средства для этого предоставляет протокол IGMP (Internet Group Management Prolocol). Когда маршрутизатор получает IP-датаграмму с групповым адресом, ему необходимо определить, находятся ли члены этой группы в локальной сети или надо пересылать датаграмму дальше. Что и выполняется при помощи протокола IGMP.

Маршрутизатор периодически посылает групповой IGMP-запрос для переучета членов группы. Это сообщение ограничено локальной подсетью, поскольку IP-параметру TTL присвоено значение 1. Каждый, принявший такое сообщение узел, запускает случайным образом установленный таймер, по истечении которого должен оправить ответ, также адресованный всей группе. При этом ответ посылает только один узел — чей таймер сработает раньше других, остальные, еще не пославшие ответ, просто выключают свои таймеры и воздерживаются от передачи. Благодаря этому сокращается объем сетевого трафика. Если маршрутизатор не получит ни одного ответа, он посчитает, что в данной подсети нет ни одного члена этой группы. Узлы, желающие войти в состав группы, немедленно отсылают этой группе сообщение и обычно повторяют его раз или два.

Протокол IGMP похож на ICMP: формально, они оба — протоколы верхнего уровня и пользуются услугами IP. На практике, протокол IGMP рассматривается и реализуется как неотъемлемая часть IP, поскольку активно влияет на его работу. Во избежание перегрузки сети, при приеме сообщения с групповым IP-адресам никогда не генерируются IGMP-сообщения об ошибках.

На рис. 3-9 показано IGMP-сообщение. Все данные, относящиеся непосредственно IGMP, являются полезной нагрузкой IP-пакета.

|  |
| --- |
| IP-заголовок |
| Версия IGMP | IGMP - тип | Не используется | Контрольная сумма |
| IGMP – групповой адрес |

Рис. IGMP-сообщение

 Поле *Версия IGMP* (IGMP Version) состоит из 4 бит и хранит одно число номер версия протокола IGMP.

Поле *IGMP-тип* (IGMP Type) состоит из 4 бит и отражает тип IGMP-сообщения. Значение 1 соответствует ответу на запрос о членстве в группе, а значение 2 — сообщению о вхождении в группу нового члена.

Поле *Не используется* (Unused) занимаете 4 бит и запланировано на будущее. При отправке его значение равно 0, а при приеме оно игнорируется.

Поле *Контрольная сумма* содержит двоичное дополнение суммы сообщения; перед началом вычислении в него заносится 0.

Поле *Групповой адрес* (Group Address) в IGMP-запросах не песет информации. При передаче оно устанавливается равным 0, а при приеме игнорируется. В IGMP-ответах оно хранит IP-адрес для группы, для которой предназначен этот ответ.

### ARP и RARP

В протоколе IPv4 каждому компьютеру, маршрутизатору и т. п. присваиваются логические адреса. С появлением технологии интерактивного видео, каждый телевизор стал потенциальным кандидатом на получение IP-адреса; то же относится к электролампочкам, кондиционерам и т. д. Но поскольку IP-адреса это лишь логические адреса, необходимо установить соответствие между ними и физическими адресами устройств, например адресами сетевых адаптеров Ethernet или TokenRing. Протокол ARP (Address Resolution Protocol) предоставляет средства для поиска физического адреса устройства при известном его IP-адресе. Этот процесс называют разрешением адреса (address resolution), Протокол ARP описан в RFC 826.

С переносом протокола IP в среду типа ATM, и RFC 2320 и 1390 был определен ARP-подобный механизм разрешения адресов в сетях, использующих технологию ATM вместо Ethernet. Для обратной процедуры — получения IP-адреса по известному физическому — используется протокол RARP (Reverse Address Resolution Protocol). Но, пожалуй, единственное полезное его применение — при загрузке бездискового компьютера. Протокол RARP описан в RFC 1931 и 903.

|  |
| --- |
| Тип оборудования (16 бит) |
| Идентификатор протокола (16 бит) |
| Длина аппаратного адреса (8 бит) | Длина IP-адреса (8 бит) |
| Операция (16 бит) |
| IP-адрес отправителя (32 бит) |
| Аппаратный адрес цели (переменная длина) |
| IP-адрес цели (32 бита) |

Рис. Формат ARP/RARP-сообщения

Формат сообщений протоколов ARP/RARP показан на рис.

Поле *Тип оборудования* (Hardware Туре) отражает тип используемого оборудовании. Например, значение 1 обозначает локальную сеть стандарта Ethernet. Поле *Идентификатор протокола* (Protocol Identifier) определяет протокол, которому принадлежит сообщение. Описаны значения только для ARP и RARP. В случае протокола ARP отправитель указывает только собственные аппаратный и IP-адрес и IP-адрес цели. Сообщения ARP могут отсылаться как широковещательно, так и направленным пакетом к шлюзу, заданному но умолчанию.

Поле *Операция* (Operation) характеризует природу сообщения. Описания значений этого поля приведены ниже.

|  |  |
| --- | --- |
| Значение | Описание |
| 1 | ARP-запрос |
| 2 | ARP-ответ |
| 3 | RARP-запрос |
| 4 | RARP-ответ |

## ТРАНСПОРТНЫЙ УРОВЕНЬ

### UDP

Протокол UDP (User Datagram Protocol) — это простейшая надстройка над IP. Он обеспечивает передачу датаграм (негарантированная доставка данных без установления логических соединении), но для непосредственной работы используется IP. Дополнительные возможности характерны ДЛЯ протокола UDP:

• уплотненно и разуплотнение капала (посредством портов UDP) — теперь сразу несколько приложении клиента, сервера или обоих могут иметь собственные потоки датаграм;

• вычисление контрольной суммы и проверка целостности принимаемых датаграм. В IP контрольная сумма затрагивает только IP-заголовок, а не все данные. В UDP контрольные суммы вычисляются па основе всего пакета, включая данные.

Протокол UDP не обеспечивает большей надежности передачи данных, чем заложена в нижележащей реализации протокола IP. Приложения, использующие UDP, должны сами повторять передачу утерянных датаграм и выявлять дублируемые, переупорядочивать датаграмы (иногда они прибывают не в том порядке, в каком были отправлены), управлять потоком данных и т. д.

Пакет протокола UDP состоит из заголовка и данных, и все это вкладывается в данные IP-пакета. Формат UDP-заголовка показам на рис

|  |  |
| --- | --- |
| Порт отправителя(16 бит) | Порт получателя(16 бит) |
| Длина | Контрольная сумма |

Рис. UDP-заголовок.

Каждое поле UDP-заголовка состоит из 16 бит. Номера портов отправителя и получателя используются при уплотнении и разуплотнении потоков для корректной доставки датаграм приложениям. В поле *Длина* (Length) указана длина в байтах UDP-пакета, включая заголовок, но без IP-заголовка и IP-параметров. Поле *Контрольная сумма* (Checksum) — это 16-разрядное двоичное дополнение суммы байт псевдозаголовка (содержащего информацию из IP-заголовка), UDP-заголовка и данных. Формат псевдозаголовка покачан на рис.

|  |
| --- |
| IP-адрес отправителя(32 бит) |
| IP-адрес получателя(32 бит) |
| 8 нулевых бит | Идентификатор протокола = 17 | UDP-длина.(16 бит) |

Рис. Псевдозаголовок UDP.

Таким образом, *Контрольная сумма* вычисляется так: в начало UDP-пакета добавляется псевдозаголовок (не физически, но логически), а затем вычисляется контрольная сумма по псевдозаголовку и UDP-пакету. Если контрольная сумма не определена, то в соответствующем поле передается значение 0. Если же реальная контрольная сумма окажется равной 0, то все биты поля будут установлены в 1.

### TCP

Сам протокол IP не гарантирует надежности передачи данных между двумя компьютерами. В свою очередь, протокол TCP (Transmission Control Protocol) обеспечивает надежный, защищенный от ошибок, полнодуплексный капал связи между двумя компьютерами. Протокол TCP использует для передачи данных протокол IP и дополнительно обрабатывает потерянные или повторяющиеся IP-датаграмы. Также TCP переупорядочивает IP-датаграмы, принятые с нарушением порядка. Другими словами, протокол TCP создаст надежный поток данных, сохраняющий порядок, заданный отправителем.

Все эти возможности TCP осуществляются путем разбиения потока данных на пакеты, которые достаточно малы, чтобы поместиться в IP-датаграмы. Эти пакеты называют сегментами (segments). При нумерации и отправке IP-датаграм применяется принцип «подтверждения приема и повторной передачи». Получатель посылает явное или неявное подтверждение каждой принятой IP-датаграмы. Отправитель ожидает некоторое время и, если не получает подтверждения, повторно отправляет IP-датаграмe.

***Заголовок ТСР-сегмента***

На рис. изображен заголовок ТСР-сегмента, часто называемый просто «ТСР-заголовок». Заголовки ТСР-сегментов имеют переменную длину, но не менее 20 байт.

|  |  |
| --- | --- |
| Порт отправителя**(16 бит)** | Порт получателя**(16 бит)** |
| Порядковый номер |
| Номер подтверждения |
| Смещение данных | Зарезервировано | Фраги | Окно |
| Контрольная сумма | Указатель на неотложные данные |
| Параметры | Выравнивание |

Рис. 3-5. Заголовок ТСР-сегмента.

Поскольку TCP-сегмента передается как полезная нагрузка IP-датаграмы, то длина данных протокола TCP вычисляется как разность между размером полезной нагрузки IP и размером ТСР-заголовка.

Протокол TCP обеспечивает падежную связь двух процессов. Конечные точки этого потока данных однозначно идентифицируются через:

• поле IP-адрес отравителя в заголовке IP-датаграмы;

• поле IP-адрес получателя в заголовке IP-датаграмы;

• поле Порт отправителя в заголовки ТСР-сегмента;

• поле Порт получателя в заголовке ТСР-сегмснта.

Поли *Порядковым номер* (Sequence number) отражает позицию первого байта данных текущего сегмента в общем потоке данных. Если установлен флаг SYN, то это поле хранит начальный порядковый помер (Initial Sequence Number, ISN), а порядковый номер первого байта данных равен ISN + 1.

Если установлен флаг АСК, то поле *Номер подтверждения* (Acknowledgement number) содержит значение, которое отправитель этого сегмента ожидает увидеть в поле *Порядковый помер* следующего сегмента (другими словам, это порядковый номер последнего успешно принятого байта данных, увеличенный на 1, в общем потоке данных). Подтверждении имеют свойство поглощаться, то есть значение поля *Номер подтверждения* отражает максимальный порядковый номер среди всех успешно принятых байт данных.

Обычно приложения, которые сразу копируют поступающие к ним данные на удаленный конец ТСP-соединения, например эмуляторы терминалов, передают всего по несколько символов и требует немедленной их отправки. Поэтому если отправитель буферизует данные, то такие приложения не очень стабильно работают. Для решения этого вопроса в протоколе TCP предусмотрена возможность принудительной передачи. Соответствующий параметр находится в 6-разрядном поле *Флаги* (Flans), значения которого перечислены в табл.

|  |  |
| --- | --- |
| Значение | Описание |
| 32 | URG: есть срочные данные |
| 16 | ACK: подтверждение приема данных |
| 8 | PSH: как можно быстрее передать данные приложению |
| 4 | RST: переустановить TCP-соединение |
| 2 | SYN: синхронизация соединения |
| 1 | FIN: конец передачи |

Поле *Смещение данных* (Data Offset) состоит из 4 бит и отражает, сколько 32-разрядных слов занимает ТСР-заголовок. Оно необходимо, поскольку ТСР-заголовок имеет переменную длину. Если требуется, заголовок дополняется нулевыми байтами.

Поле *Зарезервировано* (Reserved) состоит из 6 нулевых бит.

Поле *Окно* (Window) его значение показывает количество байт, которое готов принять, отправитель. Позицию данных, которые ожидает принять получатель, указывает порядковый номер.

Поле *Контрольная сумма* (Checksum) используется для выявления ошибок передачи. Его значение есть 16-разрядное двоичное дополнение двоичной суммы всех 16-разрядных слов заголовка и данных ТСР-сегмента, а также псевдозаголовка. Во время вычисления контрольной суммы в само поле *Контрольная сумма* заносится 0.

Псевдозаголовок TCP изображен на рис.

|  |
| --- |
| IP-адрес отправителя(32 бит) |
| IP-адрес получателя(32 бит) |
| 8 нулевых бит | Идентификатор протокола  | TCP-длина. |

Рис. Псевдозаголовок TCP.

Поле *Указатель на неотложные данные* (Urgent Pointer) заголовка ТСР-сегмента состоит из 16 бит. Его сумма с полем *Порядковый номер* дает порядковый номер байта, следующего непосредственно за срочными данными. Таким образом, реализации протокола TCP позволяют информировать приложения верхних уровней о наличии срочных данных и об их окончании.

Поле *Параметры* (Options) имеет переменную длину. Существует много различных параметров, и в ТСР-заголовке их может быть больше одного. Это поле имеет два разных формата:

• один байт - тип параметра;

• несколько байт - тин параметра, длина и значение.

Ниже описаны некоторые типы параметров.

*Конец списка параметров* (End Of Option List) обозначается типом 0, длиной 1 и не имеет числового значения. Этот параметр всегда указан последним в списке Параметров,

*Нет операции*  (No Operation) обозначается типом 1, длиной 1 и не имеет числового значении. Его используют для выравнивания следующего параметра на 32-разрядную границу.

*Максимальный размер принимаемого сегмента* (Maximum Receive Segment Size) обозначается типом 2 и длиной 4, а значение и есть максимальный размер принимаемого сегмента. Этот параметр используется отправителем для указания максимального размера ТСР-сегмента, который он способен принять; его можно задать только при создании TCP-соединения (точнее, когда установлен флаг SYN).

*Масштаб окна* (Window scale) обозначается типом 3 и длиной 3, а его значение отражает степень изменения масштаба принимающего окна, иногда этот параметр называют Сдвиг окна (Window Shift). Естественно, он используется только по договоренности обеих сторон TCP соединения.

*Разрешен SACK* (SACK Permitted) обозначается типом 3 и длиной 2. Он свидетельствует, что отправитель допускает выборочное подтверждение (SACK, Selective Acknowledgement). Этот параметр можно переслать только при установлении TCP-соединения.

Параметр *SACK* (Selective Acknowledgement) обозначается типом 5 и имеет переменную длину. Он показывает, что принято несколько блоков, но некоторые промежуточные утеряны.

Параметры *ТСР-эхо* (TCP-echo) и *TCP-эxo-ответ* (TCP-echo-reply) используются совместно для оценки времени отзыва ТСР-соединения. Параметр TCP-эхо имеет тип 6. Его можно послать только при установлении TCP-соединения. Станция, принявшая пакет с параметром ТСР-эхо должна в ответ отправить пакет с параметром ТСР-эхо-ответ (он имеет тип 7 и длину 6). Знать время отзыва достаточно важно, поскольку его занижение приводит к лишним повторам передачи.

Так же как в случае IP-заголовка, поле *Выравнивание* (Padding) — это незначащее занятое пространство, гарантирующее, что ТСР-заголовок оканчивается на 32-разрядной границе.

### SCTP

Протокол передачи с управлением потоком (Stream Control Transmission Protocol, SCTP) − это надежный транспортный протокол, который обеспечивает стабильную, упорядоченную (с сохранением порядка следования пакетов) передачу данных между двумя конечными точками (подобно TCP). Кроме того, протокол обеспечивает сохранение границ отдельных сообщений (подобно UDP). Однако в отличие от протоколов TCP и UDP протокол SCTP имеет дополнительные преимущества, такие как поддержка множественной адресации (multihoming) и многопоточности (multi-streaming) - каждая из этих возможностей увеличивает доступность узла передачи данных.

Термин «многопоточность» (англ. multi-streaming) обозначает способность SCTP параллельно передавать по нескольким независимым потокам сообщений.

Достоинства использования SCTP включают в себя:

Использование множественных интерфейсов (англ Multihoming)

Многопоточность

Поиск пути с мониторингом

Механизмы валидации и проверки подлинности

Улучшенная система контроля ошибок.

Структура пакета

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Биты  |  Биты 0–7  |  8–15  |  16–23  |  24–31 |
| +0  | Порт источника  | Порт назначения |
| 32  | Тег проверки |
| 64  | Контрольная сумма |
| 96  | Тип 1 блока  | Флаги 1 блока  | Длина 1 блока |
| 128  | Данные 1 блока |
| …  | … |
| …  | Тип N блока  | Флаги N блока  | Длина N блока |
| …  | Данные N блока |

*Порт источника:* 16 битов (целое без знака) Номер порта SCTP, используемого отправителем. Это значение может использоваться на приемной стороне для идентификации ассоциации, которой относится пакет, вместе со значениями IP-адреса отправителя, номера порта получателя и (возможно) IP-адреса получателя.

*Порт назначения:* Номер порта SCTP, в который данный пакет адресован. На приемной стороне это значение будет использоваться для демультиплексирования пакета SCTP соответствующему приложению.

*Тег проверки:* Принимающая сторона использует Тег проверки для проверки отправителя пакета SCTP. На передающей стороне значение поля Тег проверки должно устанавливаться в соответствии со значением Initiate Tag (тег верификации), полученным от партнера при создании ассоциации, за исключением перечисленных ниже случаев:

♦ В пакетах, содержащих блок Инициирования, должно быть Тег проверки = 0.

♦ В пакетах, содержащих блок SHUTDOWN-COMPLETE с установленным флагом T-bit, значение Тег проверки должно копироваться из пакета с блоком SHUTDOWN-ACK.

♦ В пакетах, содержащих блок ABORT, тег верификации может быть копией значения тега из пакета, вызвавшего передачу блока ABORT.

*Тег верификации*: 32 бита (целое без знака)

Получатель INIT (откликающаяся сторона) записывает значение параметра тег верификации. Это значение должно включаться в поле Тег проверки каждого пакета SCTP, который получатель данного блока INIT будет передавать через эту ассоциацию.

*Контрольная сумма*: 32 бита (целое без знака)

Это поле содержит контрольную сумму для данного пакета SCTP. Расчет контрольных сумм. Протокол SCTP использует алгоритм Adler-32. Контрольная сумма Adler-32 состоит из двух сумм, собранных для каждого байта: s1 представляет собой сумму значений всех

байтов, а s2 – сумму всех значений s1. Обе суммы вычисляются по модулю 65521. Сумма s1 инициализируется значением 1, s2 –

значением 0. Контрольная сумма Adler-32 сохраняется как s2\*65536 + s1 с использованием сетевого порядка байтов.

*Тип блока:* 8 битов (целое без знака) Это поле указывает тип блока, содержащегося в поле *Данные*. Поле типа может содержать значения от 0 до 254. Значение 255 зарезервировано для использования в качестве расширения.

Коды *Тип блока* выбраны таким образом, два старших бита определяли действие, которое следует предпринять конечной точке на приемной стороне если *Тип блока* не удается распознать.

00 – прекратить обработку данного пакета SCTP и отбросить его, не выполняя никаких действий по отношению к содержащейся в нем информации.

01 – прекратить обработку пакета SCTP и отбросить его без выполнения каких либо действий с содержащимися в пакете данными; указать неопознанный параметр.

10 – пропустить данный блок и продолжить обработку пакета.

11 – пропустить данный блок и продолжить обработку пакета с генерацией блока ERROR,

*Флаги*: 8 битов

Использование этих битов определяется типом блока, указанным в поле *Тип блока*. Если в документе явно не указано иное, на передающей стороне все биты следует устанавливать в 0, а на приемной - игнорировать.

*Длина*: 16 битов (целое без знака) Это поле указывает размер блока в байтах с учетом полей *Тип блока*, Флаги, Длина и Данные. Следовательно, для блоков и пустым Данные поле Length имеет значение 4. Байты заполнения в поле Длина не учитываются.

*Данные*: переменный размер Поле Данные содержит данные, передаваемые в этом блоке. Формат этого поля и способы его использования зависят от значения Тип блока.

Общий размер блока (включая поля Тип блока, Длина и Данные) должен быть кратным 4. Если размер блока не кратен 4, отправитель должен дополнить блок 1 – 3 байтами со значением 0 для выравнивания размера. Получатель должен игнорировать байты заполнения.

## СЕАНСОВЫЙ УРОВЕНЬ

### NetBIOS

Протокол NetBIOS был создан для работы в локальных сетях. Система NetBIOS предназначена для персональных ЭВМ типа IBM/PC в качестве интерфейса, независящего от фирмы-производителя. NetBIOS использует в качестве транспортных протоколов TCP и UDP. Описание NetBIOS содержится в документе (см. также RFC-1001, -1002, -1088, std-19 и std-48).

Пакет NETBIOS создан для использования группой ЭВМ, поддерживает как режим сессий (работа через соединение), так и режим дейтограмм (без установления соединения). 16-и символьные имена объектов в netbios распределяются динамически. netbios имеет собственную dns, которая может взаимодействовать с интернетовской. Имя объекта при работе с NETBIOS не может начинаться с символа \*.

Приложения могут через netbios найти нужные им ресурсы, установить связь и послать или получить информацию. NETBIOS использует для службы имен порт - 137, для службы дейтограмм - порт 138, а для сессий - порт 139.

Любая сессия начинается с netbios-запроса, задания ip-адреса и определения tcp-порта удаленного объекта, далее следует обмен NETBIOS-сообщениями, после чего сессия закрывается. Сессия осуществляет обмен информацией между двумя netbios-приложениями. Длина сообщения лежит в пределах от 0 до 131071 байт. Допустимо одновременное осуществление нескольких сессий между двумя объектами.

При организации IP-транспорта через NETBIOS IP-дейтограмма вкладывается в NETBIOS-пакет. Информационный обмен происходит в этом случае без установления связи между объектами. Имена Netbios должны содержать в себе IP-адреса. Так часть NETBIOS-адреса может иметь вид, ip.\*\*.\*\*.\*\*.\*\*, где IP указывает на тип операции (IP через Netbios), а \*\*.\*\*.\*\*.\*\* - ip-адрес. Система netbios имеет собственную систему команд (call, listen, hang up, send, receive, session status, reset, cancel, adapter status, unlink, remote program load) и примитивов для работы с дейтограммами (send datagram, send broadcast datagram, receive datagram, receive broadcast datagram). Все оконечные узлы netbios делятся на три типа:

Типы оконечных узлов NETBIOS

Широковещательные ("b") узлы;

узлы точка-точка ("p");

узлы смешанного типа ("m").

IP-адрес может ассоциироваться с одним из указанных типов. B-узлы устанавливают связь со своим партнером посредством широковещательных запросов. P- и M-узлы для этой цели используют netbios сервер имен (NBNS) и сервер распределения дейтограмм (NBDD).

NetBeui

В настоящее время (1985 г) разработана улучшенная версия протокола NETBIOS - NetBeui (NetBios extended user interface). Этот новый протокол используется операционными системами LAN manager, LAN server, Windows for Workgroups и Windows NT, а по своей функции занимает нишу протоколов TCP/IP, охватывая связной, сетевой и транспортный уровни. Здесь стандартизован формат пакетов NetBios, добавлены некоторые новые функции. netbuei базируется на протоколе OSI LLC2, вводит стандарт на формат кадра netbios (NDF) и использует NetBios в качестве интерфейса высокого уровня. Протокол обладает высоким быстродействием и служит для объединения небольших локальных сетей (20-200 ЭВМ) друг с другом или с главной ЭВМ. Этот протокол соответствует связному, сетевому и транспортному уровню модели OSI. В новых версиях NetBuei (3.0 и выше) снято ограничение на число одновременных сессий (254). Среди ограничений NetBuei следует назвать отсутствие внутренней маршрутизации и серьезные ограничения при работе в региональных сетях. По этой причине netbuei рекомендуется для локальных сетей (здесь они предпочтительнее других протоколов), а для внешних связей использовать, например, TCP/IP.

Протокол NBFCP

Для подключения терминальной системы к локальной сети или к другой терминальной системе разработан протокол NBFCP (NetBios frames control protocol, код поля протокола = 803F), который в свою очередь базируется на протоколе PPP.

Формат кадра протокола NBFCP показан на рис. 4.2.3.1.



Рис. 4.2.3.1. Формат кадра NBFCP

Поле тип содержит код 2, поле длина определяет размер заголовка, если длина=8, имя партнера отсутствует. Поле класс партнера идентифицирует тип системы отправителя (см. таблицу 4.2.3.1). Таблица возможных значений поля класс партнера приведена ниже. Поле имя партнера может иметь до 32 октетов.

Таблица 4.2.3.1 Коды класса партнера

|  |  |
| --- | --- |
| Код класса | Описание |
| 1 | Зарезервировано |
| 2 | Сервер внешнего порта PPP NetBIOS |
| 3 | Зарезервировано |
| 4 | Сервер локального доступа PPP NetBIOS |
| 5 | Зарезервировано |
| 6 | Мост PPP NetBIOS |
| 7 | Зарезервировано |
| 8 | Терминальная система PPP |

Протокол WINS

Протокол WINS разработан компанией MicroSoft для операционной среды Windows и предназначен для расширения возможностей NetBIOS.

WINS-запросы обычно транспортируются в UDP-дейтограммах. При этом используется порт отправителя=137. В поле данных размешается 2-октетное поле идентификатора, позволяющего связать запрос с откликом. Далее следует 2 байта флагов, в случае запроса туда записывается 0. За ним размещается два октета, содержащие число вопросов, 2 октета числа ответов и еще 4 нулевых октетов. Завершается кадр запроса двумя октетами поля типа (00 21 -> статус узла NetBIOS) и полем класса (для Интернет 00 01 -> (IN,1)). Такие запросы позволяют получить дополнительные данные (имя узла, его MAC-адрес, NetBIOS-имя, имя группы) об ЭВМ с заданным IP-адресом. Причем эта ЭВМ может находиться где угодно в Интернет, но непременно работать в OS Windows. Формат поля данных UDP-дейтограммы запроса показан на рис. 1. Следует иметь в виду, что имена WINS имеют ограниченное время жизни (задается параметром TTL, не путайте с полем заголовка IP) и оно по умолчанию равно 6 дням - 518400 секундам.



Рис. Формат запроса WINS

В поле данных UDP-дейтограммы отклика располагается 2-байтовое поле идентификатора, аналогичного содержащемуся в пакете запроса. Далее следует поле флагов с длиной в два октета. Формат поля данных UDP-дейтограммы отклика показан на рис. 2.



Рис. Формат отклика WINS

Поле флаги имеет следующую структуру:

|  |  |  |
| --- | --- | --- |
| 0 \_ \_ \_ | \_ \_ \_ \_ | Команда |
| \_ 000 | 0 \_ \_ \_ | Запрос |
| \_ \_ \_ \_ | \_ \_ 0 \_ | Не укорочено |
| \_ \_ \_ \_ | \_ \_ \_ 0 | Рекурсия нежелательна |
| \_ \_ \_ \_ | \_ \_ \_ 0 | Рекурсия нежелательна |

|  |  |  |
| --- | --- | --- |
| 1\_ \_ \_ | \_ \_ \_ \_ | Отклик |
| \_ 000 | 0 \_ \_ \_ | Запрос |
| \_ \_ \_ \_ | \_ \_ 0 \_ | Не укорочено |
| \_ \_ \_ \_ | \_ 1 \_ \_ | Официальны |

Для поля флаги имени характерна следующая структура:

|  |  |  |
| --- | --- | --- |
| 0\_ \_ \_ | \_ \_ \_ \_ | Уникальное имя NetBIOS |
| \_ 10 \_ | \_ \_ \_ \_ | Узел М-типа |
| \_ \_ \_ \_ | \_ 1 \_ \_ \_ | Активное имя |
| \_ \_ \_ \_ | \_ \_ 0\_ | Временное имя |

Для поля флагов имени группы характерно следующее назначение бит

|  |  |  |
| --- | --- | --- |
| 1 \_ \_ \_ | \_ \_ \_ \_ | Имя группы NetBIOS |
| \_ 10 \_ | \_ \_ \_ \_ | Узел М-типа |
| \_ \_ \_ \_ | \_ 1 \_ \_ | Активное имя |
| \_ \_ \_ \_ | \_ \_ 0\_ | Временное имя |

Протокол WINS весьма удобен для сбора данных о МАС-адресах ЭВМ в многоранговой сети, где получить эти данные с помощью ARP-запросов невозможно. Какие-то данные можно извлечь из кэша маршрутизаторов или таблиц сетевых переключателей, если они доступны с помощью SNMP-запросов. Но WINS может дать больше данных, если рабочая станция использует операционную систему Windows. Так что, когда, скажем, программа Black ICE Defender пришлет вам MAC-адрес атакера, сидящего на другом континенте, не удивляйтесь, на помощь был призван протокол WINS.

### *SOCKS*

Протокол SOCKS5 расширяет модель SOCKS v4 добавляя к ней поддержку UDP, обеспечение универсальных схем строгой аутентификации и расширяет методы адресации, добавляя поддержку доменных имен и адресов IP v6.

Реализация протокола SOCKS обычно влечет за собой перекомпиляцию или пересборку клиентских программ, работающих по протоколу TCP, для использования соответствующих функций SOCKS-библиотеки.

*Процедура для клиентов работающих по TCP*

Когда работающий по TCP клиент хочет соединиться с объектом, доступным только через файрволл, он должен открыть TCP-соединение c соответствующим SOCKS-портом SOCKS-сервера. Сервис SOCKS обычно находится на TCP-порту 1080. Если соединение прошло успешно, клиент начинает переговоры о методе аутентификации, который будет использоваться, проходит аутентификацию по выбранному методу и посылает свой запрос. SOCKS-сервер обрабатывает запрос и либо пытается установить соответствующее соединение, либо отказывает в нем.

Клиент соединяется с сервером и посылает сообщение с номером версии и выбором соответствующего метода аутентификации:

|  |  |  |
| --- | --- | --- |
| VER | NMETHODS | METHODS |
| 1 | 1 | 1 to 255 |

Значение поля VER равно X'05' для данной версии протокола. Поле NMETHODS содержит число октетов в идентификаторах методов авторизации в поле METHODS.

Сервер выбирает один из предложенных методов, перечисленных в METHODS, и посылает ответ о выбранном методе:

|  |  |
| --- | --- |
| VER | METHODS |
| 1 | 1 |

Если выбранный метод в METHOD равен X'FF', то ни один из предложенных клиентом методов не применим и клиент должен закрыть соединение.

Эти значения определены для поля METHOD.

Затем клиент и сервер начинают аутентификацию согласно выбранному методу.

Совместимые реализации должны поддерживать GSSAPI и могут поддерживать аутентификацию USERNAME/PASSWORD.

После того как аутентификация выполнена, клиент посылает детали запроса. Если выбранный метод аутентификации требует особое формирование пакетов с целью проверки целостности и/или конфиденциальности, запросы должны инкапсулироваться в пакет, формат которого определяется выбранным методом.

SOCKS-запрос формируется следующим образом:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VER | CMD | RSV | ATYP | DST.ADDR | DST.PORT |
| 1 | 1 | X'00' | 1 | Variable | 2 |

Где:

VER версия протокола: X'05'

CMD CONNECT X'01'; BIND X'02'; UDP ASSOCIATE X'03'

RSV зарезервировано

ATYP тип адреса

DST.ADDR требуемый адрес

DST.PORT требуемый порт (в сетевом порядке октетов)

SOCKS-сервер обрабатывает запрос на основании исходного и целевого адресов и посылает одно или несколько сообщений в ответ, в соответствии с типом запроса.

SOCKS-запрос посылается клиентом как только он установил соединение с SOCKS-сервером и выполнил аутентификацию. Сервер обрабатывает запрос и посылает ответ в следующей форме:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ER | REP | RSV | ATYP | BND.ADDR | BND.PORT |
| 1 | 1 | X'00' | 1 | Variable | 2 |

VER версия протокола: X'05'

REP код ответа:

RSV зарезервирован

ATYP тип последующего адреса

BND.ADDR выданный сервером адрес

BND.PORT выданный сервером порт (в сетевом порядке октетов)

Значения зарезервированных (RSV) полей должны быть установлены в X'00'.

Если выбранный метод аутентификации требует особое формирование пакетов с целью проверки целостности и/или конфиденциальности, запросы должны инкапсулироваться в пакет, формат которого определяется выбранным методом.

*Процедура для клиентов работающих по UDP*

Клиент, работающий по UDP, должен посылать свои датаграммы на порт пересылающего их UDP-сервера, указанного в поле BND.PORT в ответе на запрос UDP ASSOCIATE. Если выбранная схема аутентификации требует особое формирование пакетов с целью проверки целостности и/или конфиденциальности, датаграмма должна инкапсулироваться в пакет, формат которого определяется выбранной схемой. Каждая UDP-датаграмма содержит в себе заголовок UDP-запроса:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RSV | FRAG | ATYP | DST.ADDR | DST.PORT | DATA |
| 2 | 1 | 1 | Variable | 2 | Variable |

Поля заголовка UDP-запроса:

\* RSV зарезервировано X'0000'

\* FRAG текущий номер фрагмента

\* ATYP тип адреса:

\* DST.ADDR требуемый целевой адрес

\* DST.PORT требуемый целевой порт

\* DATA пользовательские данные

Когда пересылающий UDP-датаграммы сервер пересылает датаграмму, он делает это молча, без какого-либо уведомления выполнившего запрос клиента. Аналогично, сервер будет молча отбрасывать датаграмы, которые он не может или не будет пересылать. Когда пересылающий UDP-датаграмы сервер получает ответную датаграмму с удаленного хоста, он должен инкапсулировать эту датаграмму используя помимо заголовка UDP-запроса еще и инкапсуляцию, определяемую выбранной схемой аутентификации.

Обращение к пересылающему UDP-датаграммы серверу должно производиться с ожидаемого SOCKS-сервером IP-адреса клиента, который (клиент) будет посылать датаграммы на BND.PORT, данный в ответе на UDP ASSOCIATE. Сервер должен отбрасывать датаграммы полученные с любого IP-адреса, отличного от того, что был записан для этой связи.

Поле FRAG показывает, является ли эта датаграмма самостоятельной или же фрагментом. Если датаграмма - фрагмент, то установленный старший бит является признаком последнего фрагмента, в то время как значение X'00' показывает, что это обычная датаграмма. Значения от 1 до 127 обозначают на позицию фрагменте в последовательности. Каждый получатель будет иметь REASSEMBLY QUEUE (очередь сборки) и REASSEMBLY TIMER (таймер сборки) связанные с такой фрагментной датаграммой. Очередь сборки должна быть пере инициализирована и связанные с ней фрагменты выкинуты всякий раз при истечении таймера сборки или с приходом новой датаграммы, чье значение в поле FRAG меньше, чем наибольшее значение поля FRAG датаграмм, обработанных при сборке фрагмента. Таймер сборки должен быть не менее 5 секунд. Приложениям рекомендуется избегать фрагментацию везде, где только это возможно.

Реализация фрагментации опциональна, в реализациях где фрагментация не поддерживается, должны отбрасываться любые датаграммы, у которых поле FRAG отлично от X'00'.

Программный интерфейс для UDP работающего через SOCKS должен сообщать о оставшемся свободном пространстве в буфере для UDP-датаграммы, которое меньше, чем действительный размер буфера, выделенный операционной системой

Так как в заголовке UDP-запроса, включенного в датаграмму, нет информации о длине данных, то приложение должно помнить об этом самостоятельно.

### RPC

Remote Procedure Call (RPC) - Вызов удаленных процедур.

Идея вызова удаленных процедур (Remote Procedure Call - RPC) состоит в расширении механизма передачи управления и данных внутри программы, выполняющейся на одной машине, на передачу управления и данных через сеть. Средства удаленного вызова процедур предназначены для облегчения организации распределенных вычислений. Наибольшая эффективность использования RPC достигается в тех приложениях, в которых существует интерактивная связь между удаленными компонентами с небольшим временем ответов и относительно малым количеством передаваемых данных. Такие приложения называются RPC-ориентированными.

Характерными чертами вызова локальных процедур являются:

1. Асимметричность, то есть одна из взаимодействующих сторон является инициатором;
2. Синхронность, то есть выполнение вызывающей процедуры при останавливается с момента выдачи запроса и возобновляется только после возврата из вызываемой процедуры.

Реализация удаленных вызовов существенно сложнее реализации вызовов локальных процедур, так как RPC не может рассчитывать на разделяемую память. Это означает, что параметры RPC не должны содержать указателей на ячейки нестековой памяти и что значения параметров должны копироваться с одного компьютера на другой. Отличием RPC является то, что он использует нижележащую систему связи. Кроме того, существует ряд проблем, связанных с неоднородностью языков программирования и операционных сред.

На рисунке показана последовательность команд, которую необходимо выполнить для каждого RPC-вызова.



Программные реализации системы, как правило, поддерживают один или два протокола. Например, система RPC разработки фирмы Sun Microsystems поддерживает передачу сообщений с использованием протоколов TCP и UDP. Выбор того или иного протокола зависит от требований приложения. Выбор протокола UDP оправдан для приложений, обладающих следующими характеристиками:

* Вызываемые процедуры идемпотентны
* Размер передаваемых аргументов и возвращаемого результата меньше размера пакета UDP - 8 Кбайт.
* Сервер обеспечивает работу с несколькими сотнями клиентов. Поскольку при работе с протоколами TCP сервер вынужден поддерживать соединение с каждым из активных клиентов, это занимает значительную часть его ресурсов. Протокол UDP в этом отношении является менее ресурсоемким

С другой стороны, TCP обеспечивает эффективную работу приложений со следующими характеристиками:

* Приложению требуется надежный протокол передачи
* Вызываемые процедуры неидемпонентны
* Размер аргументов или возвращаемого результата превышает 8 Кбайт

Выбор протокола обычно остается за клиентом, и система по-разному организует формирование и передачу сообщений. Так, при использовании протокола TCP, для которого передаваемые данные представляют собой поток байтов, необходимо отделить сообщения друг от друга. Для этого например, применяется протокол маркировки записей, описанный в RFC1057 "RPC: Remote Procedure Call Protocol specification version 2", при котором в начале каждого сообщения помещается 32-разрядное целое число, определяющее размер сообщения в байтах.

По-разному обстоит дело и с семантикой вызова. Например, если RPC выполняется с использованием ненадежного транспортного протокола (UDP), система выполняет повторную передачу сообщения через короткие промежутки времени (тайм-ауты). Если приложение-клиент не получает отклик, то с уверенностью можно сказать, что процедура была выполнена ноль или большее число раз. Если отклик был получен, приложение может сделать вывод, что процедура была выполнена хотя бы однажды. При использовании надежного транспортного протокола (TCP) в случае получения отклика можно сказать, что процедура была выполнена один раз. Если же отклик не получен, определенно сказать, что процедура выполнена не была, нельзя.

Существуют множество технологий, обеспечивающих RPC:

* Sun RPC (бинарный протокол на базе TCP и UDP и XDR) RFC-1831 второе название ONC RPC RFC-1833
* .NET Remoting (бинарный протокол на базе TCP, UDP, HTTP)
* SOAP — Simple Object Access Protocol (текстовый протокол на базе HTTP) см. спецификацию: RFC-4227
* XML RPC (текстовый протокол на базе HTTP) см. спецификацию: RFC-3529
* Java RMI — Java Remote Method Invocation — см. спецификацию: http://java.sun.com/j2se/1.5.0/docs/guide/rmi/index.html
* DCOM — Distributed Component Object Model известный как MSRPC Microsoft Remote Procedure Call или «Network OLE» (объектно-ориентированное расширение DCE RPC, позволяющее передавать ссылки на объекты и вызывать методы объектов через таковые ссылки).

## ПРЕДСТАВИТЕЛЬСКИЙ УРОВЕНЬ

### X.25

X.25 - протокол (ISO 8208:1989; RFC-887, -1381, -1382, -1461, -1598, -1613), который определяет синхронный интерфейс между терминальным оборудованием (DTE - Data Terminal Equipment) и оборудованием передачи данных (DCE - Data Communication Equipment) для терминалов, работающих в пакетном режиме. Главный недостаток протокола X.25 - большие задержки отклика (типовое значение 0.6 сек).Соединение DTE - DTE осуществляется через DCE. В протоколе X.25 DCE и DTE используют статистическое мультиплексирование с делением по времени. Одновременно могут реализовываться несколько обменных процессов. Схема взаимодействия DTE и DCE выглядит как:

DTE - <логический канал> - DCE <виртуальное соединение> - DCE - <логический канал> - DTE

Сетевой адрес пользователя состоит из 12 десятичных цифр. 1-4 - идентификатор сети передачи данных (3 - страна, 4 - сеть); 5-12 - национальный номер (5-7 местная область, 8-12 - местный номер). Международная система адресации для систем передачи данных общего пользователя описана в рекомендациях X.121 международного комитета по телефонии и телеграфии. Каждое подключение к сети коммутации пакетов имеет свой национальный номер. Протокол X.25 не определяет технику маршрутизации пакетов по сети. Для целей управления в сетях X.25 используется протокол snmp и база данных MIB (как и в сетях Интернет). Три базовых уровня протокола X.25 и схема потоков информации отображены на рис.

Саму технологию можно представить в виде взаимодействия следующих уровней:

* 1. физический x.21 (X.21bis).
	2. канальный (HDLC - high data link communication - протокол высокого уровня управления каналом). Этот уровень и последующие реализуются программным образом.
	3. сетевой (пакетный).

X.21 - универсальный интерфейс между оконечным оборудованием (DTE) и аппаратурой передачи данных (DCE) для синхронного режима работы в сетях общего пользования. X.21bis - тоже, но для модемов, удовлетворяющих рекомендациям серии V. Для канального уровня используется подмножество протокола HDLC (являющегося развитием стандарта SDLC IBM), обеспечивающее возможность автоматической повторной передачи в случае возникновения ошибок в линии.



Формат кадра для протокола HDLC показан на рис.



Открывающий и закрывающий флаги для бит-ориентированного формата несут в себе код 0x7e. Когда не передается никакой информации, по каналу пересылается непрерывный поток флагов 01111110. Посылка более 6 единиц подряд воспринимается как флаг абортирования связи. Если необходимо передать информационную последовательность 01111110, после первых пяти единиц вводится дополнительный нуль, приемник восстанавливает истинную информацию, удаляя эти лишние нули. В случае байт-ориентированных кадров открывающий и завершающий флаги имеют по два байта. Адрес в пакете X.25 занимает всего один байт, что определяет предельное число терминальных устройств, подключаемых к одному каналу. Кадр на уровне 2 имеет двухбайтовый заголовок, содержащий байт адреса и байт типа. Для нумерации кадров на уровне 2 используется 3 бита. При работе со скользящим окном откликов это позволяет иметь до 7 кадров в очереди. При использовании спутниковых каналов с большими задержками можно переходить в режим расширенной нумерации (7 бит), где длина очереди может достигать 128. Если удаленный партнер не способен работать в режиме расширенной нумерации, он отклонит запрос соединения. При работе в режиме расширенной нумерации возможно применение 3-байтовых заголовков вместо двухбайтовых.

Значения *поля идентификатора общего формата* (GFI - general format identifier) приведено в таблице. Бит 8 этого поля (Q) используется в информационных пакетах как индикатор уровня передаваемых данных. *Групповой номер логического канала* и *номер логического канала* присваиваются по соглашению с администрацией сети во время постановки на обслуживание. Поля групповой номер логического канала и номер логического канала присутствуют во всех пакетах кроме пакетов регистрации и повторного пуска, где они принимают нулевое значение.

Таблица Коды идентификатора общего формата GFI

|  |  |  |
| --- | --- | --- |
| Тип пакета | Модуль нумерации | Номера битов |
|  |  | 8 | 7 | 6 | 5 |
| Установка соединения | 8128 | 00 | xx | 01 | 10 |
| Разрыв соединения, управление потоком, повторный пуск, регистрация, диагностика | 8128 | 00 | 00 | 01 | 10 |
| Данные | 8128 | xx | xx | 01 | 10 |
| Расширение | - | 0 | 0 | 1 | 1 |

x - бит может принимать значения 0 или 1.

Допустимые значения кодов в поле *тип пакета* приведены в таблице

|  |  |
| --- | --- |
| **Тип пакета** | **Октет 3** |
| Биты | 8 7 6 5 4 3 2 1 |
| Запрос | 0 0 0 0 1 0 1 1 |
| Запрос принят | 0 0 0 0 1 1 1 1 |
| Запрос завершения | 0 0 0 1 0 0 1 1 |
| Подтверждение завершения | 0 0 0 1 0 1 1 1 |
| Данные | x x x x x x x 0 |
| Прерывание | 0 0 1 0 0 0 1 1 |
| Подтверждение прерывания  | 0 0 1 0 0 1 1 1 |
| Готовность к приему по модулю 8 (RR) | x x x 0 0 0 0 1 |
| Готовность к приему по модулю 128 (RR) | 0 0 0 0 0 0 0 1 |
| Неготовность к приему по модулю 8 (RNR) | x x x 0 0 1 0 1 |
| Неготовность к приему по модулю 128 (RNR) | 0 0 0 0 0 1 0 1 |
| Запрос повторной установки | 0 0 0 1 1 0 1 1 |
| Подтверждение повторной установки | 0 0 0 1 1 1 1 1 |
| Запрос повторного пуска | 1 1 1 1 1 0 1 1  |
| Подтверждение повторного пуска | 1 1 1 1 1 1 1 1 |
| Диагностика | 1 1 1 1 0 0 0 1 |
| Запрос регистрации | 1 1 1 1 0 0 1 1 |
| Подтверждение регистрации | 1 1 1 1 0 1 1 1 |

x - отмечет разряды, которые могут принимать значения 0 или 1.

Четырехбитовые поля *длина адреса отправителя* и *длина адреса получателя* характеризуют длины последующих полей переменной длины. Длина выражается в полуоктетах. Далее следуют соответствующие адреса. В каждом полуоктете записывается десятичная цифра адреса, при необходимости поле адреса дополняется нулями до целого числа октетов.

Для реализации работы сетей ISDN по существующим каналам сети X.25 разработан протокол X.31. X.31 организует канал пользователь-маршрутизатор X.25 (через посредство ISDN) и регламентирует работу ISDN с пакетами X.25.

Для решения первой задачи используется сообщение SETUP. Вторая задача решается, когда канал до маршрутизатора сформирован. На этом этапе привлекается набор протоколов X.25, возможно применение протокола X.75 (ISO 8208), который является расширением X.25 для межсетевых связей.

Independent Computing Architecture (ICA) это закрытый протокол для сервера приложений, разработанного компанией Citrix Systems. Протокол определяет спецификацию обмена данными между сервером и клиентами, но не встроен ни в одну из платформ.

External Data Representation (XDR) — международный стандарт передачи данных в Интернете, используемый в различных RFC для описания типов. Он позволяет данным быть упакованными не зависящим от архитектуры способом, таким образом, данные могут передаваться между гетерогенными компьютерными системами. ,

NetWare Core Protocol (NCP) — это сетевой протокол, который используется в некоторых продуктах от Novell, является надстройкой над протоколом IPX или TCP/IP и используется для организации обмена между рабочей станцией и файловым сервером. В основном NCP связан и используется в операционной системе NetWare, но его части были реализованы на другие платформы, такие как Linux, Windows NT и Unix.

## ПРИКЛАДНОЙ УРОВЕНЬ

### HTTP

Протокол HTTP - это протокол запросов/ответов. Клиент посылает серверу запрос, содержащий метод запроса, URI, версию протокола, MIME-подобное сообщение, содержащее модификаторы запроса, клиентскую информацию, и, возможно, тело запроса, по соединению. Сервер отвечает строкой состояния, включающей версию протокола сообщения, код успешного выполнения или код ошибки, MIME-подобное сообщение, содержащее информацию о сервере, метаинформацию объекта, и, возможно, тело объекта.

Большинство HTTP соединений инициализируется агентом пользователя и состоит из запроса, который нужно применить к ресурсу на некотором первоначальном сервере. В самом простом случае, он может быть выполнен посредством одиночного соединения (v) между агентом пользователя (UA) и первоначальным сервером (O).

цепочка запросов --------------------->

UA -------------------v------------------- O

<----------------------- цепочка ответов

Более сложная ситуация возникает, когда в цепочке запросов/ответов присутствует один или несколько посредников. Существуют три основных разновидности посредников: прокси-сервера, шлюзы, и туннели. Прокси-сервер является агентом-посредником, который получает запросы на некоторый URI в абсолютной форме, изменяет все сообщение или его часть, и отсылает измененный запрос серверу, идентифицированному URI. Шлюз - это принимающий агент, действующий как бы уровень выше некоторого другого сервера(ов) и, в случае необходимости, транслирующий запросы в протокол основного сервера. Туннель действует как реле между двумя соединениями не изменяя сообщения; туннели используются, когда связь нужно производить через посредника (например Firewall), который не понимает содержание сообщений.

цепочка запросов ----------------------------------->

UA -----v----- A -----v----- B -----v----- C -----v----- O

<------------------------------------ цепочка ответов

На последнем рисунке показаны три посредника (A, B, и C) между агентом пользователя и первоначальным сервером. Запросы и ответы передаются через четыре отдельных соединения. Это различие важно, так как некоторые опции HTTP соединения применимы только к соединению с ближайшим не туннельным соседом, некоторые только к конечным точкам цепочки, а некоторые ко всем соединениям в цепочке. Хотя диаграмма линейна, каждый участник может быть задействован в нескольких соединениях одновременно.

Любая сторона соединения, которая не действует как туннель, может использовать внутренний кэш для обработки запросов. Эффект кэша заключается в том, что цепочка запросов/ответов сокращается, если один из участников в цепочке имеет кэшированный ответ, применимый к данному запросу.

HTTP соединение обычно происходит посредством TCP/IP соединений. Заданный по умолчанию порт TCP - 80, но могут использоваться и другие порты. HTTP также может быть реализован посредством любого другого протокола Интернета, или других сетей. HTTP необходима только надежная передача данных, следовательно может использоваться любой протокол, который гарантирует надежную передачу данных; отображение структуры запроса и ответа HTTP/1.1 на транспортные модули данных рассматриваемого протокола.

URI известны под многими именами: WWW адреса, Универсальные Идентификаторы Документов, Универсальные Идентификаторы Ресурсов (URI), и, в заключение, как комбинация Единообразных Идентификаторов Ресурса (Uniform Resource Locators, URL) и Единообразных Имен Ресурса (Uniform Resource Names, URN). HTTP определяет URL просто как строку определенного формата, которая идентифицирует – через имя, расположение, или любую другую характеристику - ресурс.

Описания протокола HTTP, содержащиеся в документах RFC 1945 и RFC 2616, определяют формат сообщений, предназначенных для обмена между клиентом и сервером. В HTTP существуют два типа сообщений: запросы и ответы.

***Сообщение-запрос***

Типичное сообщение-запрос протокола HTTP выглядит следующим образом:

*GET /somedir/page.html HTTP/1.1*

*Host:* [*www.someschool.edu*](http://www.someschool.edu)

*Connection: close*

*User-agent: Mozilla/4.0*

*Accept-language: fr*

Сообщение состоит из пяти строк, каждая из которых оканчивается парой символов для перехода на новую строку, а последняя строка — дополнительной парой указанных символов. В общем случае число строк сообщения может быть как больше, так и меньше пяти (вплоть до одной строки). Первая строка называется строкой запроса, а следующие строки — строками заголовка. Строка запроса содержит три поля: *поле метода*, *поле URL* и *поле версии HTTP*. *Поле метода* может принимать различные значения, например GET, POST и HEAD. Метод GET применяется в случаях, когда требуемый объект идентифицируется URL-адресом. Приведенное сообщение содержит URL-адрес */somedir/page.html*. Поле версии HTTP не требует дополнительных.

Теперь рассмотрим *строки заголовка*. Строка Host: [*www.someschool.edu*](http://www.someschool.edu) содержит адрес хоста, на котором находится объект. С помощью строки Connection: close браузер сообщает серверу о том, что не следует использовать постоянное соединение, и установленное TCP-соединение должно быть закрыто сразу после передачи тре­буемого объекта. При этом браузер поддерживает версию 1.1 протокола HTTP. В строке The User-agent: указан агент пользователя, то есть тип браузера, сгенерировавшего запрос. В данном случае это браузер Mozilla фирмы Netscape. Строка User-agent: является весьма полезной, поскольку на сервере могут храниться несколько версий одного документа, предназначенных для разных браузеров и адресуемых одним URL-адресом. Наконец, строка Accept-language: указывает на то, что пользователю по возможности должна быть выслана версия документа на французском языке (в случае ее наличия на сервере); в противном случае будет выслана версия документа на языке, заданном по умолчанию. Строка Accept-language: является одной из множества заголовочных строк согласования данных.

Рассмотрев конкретный пример, обратимся теперь к общему формату запроса, представленному на рис. 4.6. Как можно видеть, пример вполне соответствует этому формату; тем не менее после строк заголовка и пустой строки формат сообщения предусматривает наличие тела сообщения. Тело сообщения остается пустым при использовании метода GET и заполняется при использовании метода POST. Метод POST применяется в случаях, когда пользователь заполняет формы, на­пример, вводит слово для поиска в поисковой системе. Заполнение форм приводит к генерации запроса, а содержимое web-страницы зависит от данных, введенных в формы. Итак, если поле метода содержит значение POST, то в теле сообщения находятся данные, введенные в формы.



Рис. 4.6. Общий формат сообщения-запроса

Необходимо отметить, что в запросах, создаваемых с помощью форм, не всегда применяется метод POST. Напротив, HTML-формы часто используют метод GET и подставляют введенные значения в URL-адрес требуемой страницы. К примеру, если пользователь ввел в формы два значения, monkeys и bananas, то запрашива­емый с помощью метода GET URL-адрес будет иметь вид [*www.somesite.com/ani-*](http://www.somesite.com/ani-)*malsearch?monkeys&bananas*. Вполне вероятно, что вы нередко встречали подобные конструкции, путешествуя в web.

Метод HEAD схож с методом GET. При получении запроса с методом HEAD сер­вер формирует ответ, однако не осуществляет пересылку объекта. Разработчики приложений часто используют метод HEAD для отладки ошибок.

В спецификации НТТР/1.0 указаны лишь три метода: GET, POST и HEAD. Спе­цификация НТТР/1.1 располагает более широким набором методов, в который, кроме перечисленных выше, входят PUT и DELETE. Метод PUT часто применяется в средствах web-публикаций и позволяет поместить объект с заданным URL-адресом на web-сервер, а метод DELETE — удалить объект, расположенный на web-сервере.

***Сообщение-ответ***

Структура данного сообщения состоит из трех частей: *строки состояния*, шести *строк заголовка* и *тела сообщения*. *Тело сообщения* содержит требуемый объект. Строка состояния образована из трех полей: поля версии протокола, поля кода состояния и поля соответствующей коду информации, описывающей это состояние.

Сервер использует строку *Connection: close* для уведомления клиента о том, что TCP-соединение будет закрыто после того, как окончится пересылка объекта. Строка *The Date:* содержит дату и время создания ответа. Эта дата указывает момент извлечения объекта из места его хранения и включения в тело сообщения. Строка *Server:* говорит о том, что сообщение было создано сервером Apache, и она аналогична строке User-agent: в сообщении-запросе. Строка *Last-Modified:* содержит дату и время создания или последнего изменения объекта. Содержимое строки Last-Modified: крайне важно для кэширования объектов как на локальных клиентах, так и на сетевых кэш-серверах (часто называемых прокси-серверами). Строка *Content-Length:* содержит размер пересылаемого объекта в байтах, а строка *Content-Type:* указывает на то, что объект является текстом в формате HTML.

*НТТР/1.1 200 ОК Connection: close*

*Date: Thu. 06 Aug 2012 12:00:15 GMT*

*Server: Apache/1.3.0 (Unix)*

*Last-Modified: Mon. 22 Jun 2012 09:23:24 GMT*

*Content-Length: 6821*

*Content-Type: text/html*

*(data data data data data ...)*

Если сервер получает запрос, в котором указана версия НТТР 1.0, постоянное соединение не будет использоваться, даже при поддержке сервером протокола НТТР 1.1. Это необходимо потому, что спецификация HTTP 1.0 не предусматривает постоянных соединений.

Рассмотрев частный случай, обратимся к общему формату ответного сообщения, представленному на рис.



Рис Общий формат сообщения-ответа

Поля кода состояния и информации о состоянии взаимосвязаны и фактически отражают результат обработки запроса. Ниже приведены несколько наиболее часто встречающихся пар, содержащих код состояния и информацию об этом состоянии.

*200 OK:*

*Запрос успешно обработан, объект получен и включен в ответ.*

*301 Moved Permanently:*

*Объект был перемещен; новый URL-адрес указан в строке ответа Location:. Программа клиента автоматически выполнит запрос по новому адресу.*

*400 Bad Request:*

*Общая ошибка, вызванная невозможностью интерпретации запроса сервером.*

*404 Not Found:*

*Запрашиваемый документ не найден на сервере.*

*505 HTTP Version Not Supported:*

*Указанная в запросе версия HTTP не поддерживается сервером.*

Для браузера включаемые строки зависят от фирмы-разработчика, версии продукта (например, браузер, разработанный для спецификации НТТР 1.0, не сможет генерировать строки, характерные для спецификации НТТР 1.1), пользовательских настроек (например, языка), наличия на компьютере версии (возможно, устаревшей) запрашиваемого объекта. Аналогичная ситуация характерна для серверов: существуют различные программные продукты, их версии, настройки, совместно влияющие на включаемые строки заголовка.

**Взаимодействие пользователя с сервером**

HTTP-сервер не запоминает информацию о состоянии соединения. Это упрощает разработку сервера и позволяет достичь значительной про­изводительности за счет одновременного обслуживания сотен ТСР-соединений. Тем не менее возможность распознавания пользователей сервером является весьма желательной. Причиной этому может служить необходимость разграничения прав доступа к информации, находящейся на сервере, либо предоставление каждому пользователю собственного набора информационных услуг. Протокол HTTP предусматривает два механизма идентификации пользователей: авторизацию и объекты cookie.

***Авторизация***

Вероятно, вам приходилось сталкиваться с ситуациями, когда сервер предлагал вам ввести имя пользователя и пароль для доступа к своему информационному пространству. Подобный механизм доступа называется авторизацией. Запрос и получение авторизации в HTTP зачастую производятся с помощью особых заголовков и кодов состояния. Рассмотрим следующий пример. Пусть клиент инициирует запрос объекта, причем объект находится на сервере, требующем авторизации.

Сначала клиент формирует обычный запрос, не содержащий специальных строк. Сервер возвращает ответ с пустым телом и кодом состояния 401 Authorization Required. В специальной строке WWW-Authenticate: заголовка содержится описание метода проведения авторизации; как правило, это описание указывает на необхо­димость ввести имя пользователя и пароль.

Получив подобное сообщение, клиент запрашивает имя пользователя и пароль. По окончании ввода генерируется новый запрос, содержащий строку Authorization: с введенными пользователем данными. Получив первый объект, клиент продолжает отсылать серверу имя пользователя и пароль для всех остальных запрашиваемых объектов. Обычно это происходит до тех пор, пока работа клиента не будет завершена пользователем. Таким образом, в течение всего сеанса с клиентом имя пользователя и пароль кэшируются и не запрашиваются у пользователя многократно.

***Cookie***

Объекты cookie являются альтернативным авторизации средством идентифика­ции пользователей. Описание cookie находится в документе RFC 2109. Обычно объекты cookie находят применение в Интернет-порталах (например, Yahoo!), электронной коммерции (например, Amazon) и рекламе (например, DoubleClick).

Технология cookie подразумевает наличие четырех основных компонентов:

1. заголовочной cookie-строки в ответном сообщении сервера;
2. заголовочной cookie-строки в запросе клиента;
3. cookie-файла, находящегося на стороне клиента и обрабатываемого браузером;
4. удаленной базы данных, расположенной на web-сайте.

Рассмотрим типичный пример использования объекта cookie. Предположим, пользователь применяет для доступа в web один и тот же браузер (пусть это будет Internet Explorer) и впервые оказывается на сайте, предоставляющем услуги элек­тронной коммерции. Доступ к сайту осуществляется при помощи технологии cookie. При первом доступе сервер генерирует уникальный идентификационный номер для пользователя, создает в своей базе данных запись с индексом, равным идентификационному номеру, и отсылает клиенту пользователя ответное сообщение, включающее специальную строку Set-cookie: заголовка, содержащую идентификационный номер.

Получив ответ, браузер анализирует его и добавляет строку Set-cookie: в cookie-файл. Этот файл содержит имена хостов и соответствующие идентификационные номера серверов. Каждый раз при формировании запроса к web-сайту браузер обращается к cookie-файлу, извлекает из него нужный идентификационный номер, включает его в запрос и отсылает серверу.

Таким образом, сервер может собирать информацию о деятельности у себя пользователя: времени доступа, посещенных страницах и т. п. Это, в свою очередь, позволяет серверу организовать «карту покупателя» со списком сделанных во время сеанса покупок и дает возможность сразу оплатить их.

При повторных входах на сайт идентификационный номер снова будет передан серверу. Сервер, располагающий информацией о предыдущих покупках пользователя, может на ее основе составить рекомендации о новых покупках. Зачастую подобные коммерческие сайты позволяют пользователям пройти регистрацию, указав свои имя, фамилию, почтовый адрес, номер кредитной карты, адрес электронного почтового ящика и др. Введенные данные заносятся в базу данных серве­ра. Именно с помощью описанного механизма осуществляется «покупка одним щелчком мыши» — потребность ввода личных данных отпадает.

Итак, объекты cookie представляют собой средство идентификации пользователей. При первом сеансе доступа пользователь вводит какой-либо идентификационный параметр (например, свое имя), а сервер в ответном сообщении отсылает пользователю заголовочную cookie-строку, идентифицируя его. Кроме того, технология cookie позволяет создать подобие дополнительного «сеансового уровня» для протокола HTTP, не запоминающего информацию о соединении. К примеру, когда пользователь подключается к web-приложению электронной почты, браузер отсылает cookie-информацию серверу, позволяющую идентифицировать пользователя во время сеанса.

Несмотря на то, что объекты cookie позволяют упростить процесс покупок для пользователя, они, как и приведенная выше схема авторизации, весьма ненадежны с точки зрения обеспечения конфиденциальности информации. Процедура регистрации приводит к выдаче пользователем данных личного характера, которые при использовании небезопасного cookie-доступа могут стать известны дру­гим людям. Кроме того, объекты cookie могут применяться для сбора информации о поведении пользователя на множестве web-сайтов. Web-страницы, содержащие бан-нерную рекламу, прибегают к cookie для получения объектов баннерной рекламы (как правило, рисунков в формате JPEG и GIF) с серверов рекламного агентства. Каждый из запросов к серверу рекламного агентства может содержать объект cookie, генерируемый сайтом рекламного агентства. Поскольку агентства обычно связаны с множеством web-страниц, это позволяет им отслеживать пути отдельных пользователей в web и анализировать собранную информацию.

### FTP и TFTP

Протокол FTP (File Transfer Protocol) —относительно безопасный способ перемещения файлов между различными компьютерами. В качестве транспортного механизма для передачи данных FTP применяет протокол TCP. Его средствами пользователь может предъявлять «удостоверения личности» серверу, а затем просматривать папки и передавать файлы в обоих направлениях. FTP позволяет передавать данные как между клиентом и FTP-сервером, так и между двумя удаленными компьютерами.

Спецификация RFC 959 определяет FTP-модель в терминах модели «пользователь — сервер».

FTP поддерживает две логические связи между компьютерами. Одна связь служит для удаленного доступа и использует протокол Telnet. Другая связь предназначена для обмена данными. Сервер производит операцию passive open для порта 21 и ждет соединения с клиентом. Клиент осуществляет операцию active open для порта 21. Канал остается активным до завершения процедуры FTP. TOS (тип IP-сервиса) соответствует минимуму задержки, так как этот канал используется для ручного ввода команд. Канал для передачи данных (TCP) формируется каждый раз для пересылки файлов. Канал открывается перед началом пересылки и закрывается по коду end\_of\_file (конец файла). IP-тип сервиса (TOS) в этом случае ориентирован на максимальную пропускную способность.

Конечный пользователь взаимодействует с протокольным интерпретатором, в задачи которого входит управление обменом информацией между пользователем и файловой системой, как местной, так и удаленной.

Сначала по запросу клиента формируется канал управления, который в дальнейшем используется для передачи команд от клиента и откликов от сервера. Информационный канал формируется сервером по команде клиента, он не должен существовать постоянно на протяжении всей FTP-сессии и может формироваться и ликвидироваться по мере необходимости. Канал управления может быть закрыт только после завершения информационного обмена. Для канала управления используется протокол **Telnet**. После того как управляющий канал сформирован, клиент может посылать по нему команды. Сервер воспринимает, интерпретирует эти команды и передает отклики.

Возможна и другая схема взаимодействия, когда по инициативе клиента осуществляется файловый обмен между двумя ЭВМ, ни одна из которых не является машиной клиента

На фазе задания режима обмена предоставляются следующие возможности:

Команда Block сохраняет структуру логических записей файла.

Команда Stream устанавливает режим, при котором не производится пересылки контрольной информации для блоков. Это наиболее быстрый режим обмена, он работает по умолчанию.

Команда TYPE может задать режимы обмена

• NVT-ASCII;

• EBCDIC — используется в основном на мэйнфреймах;

• IMAGE — описывает представление двоичных исполняемых файлов в виде последовательностей 8-разрядпых байтов, или октетов;

• LOCAL — описывает представление двоичных исполняемых файлов с указанным размером байта.

Для копирования файла из удаленного сервера используется команда GET, для копирования группы файлов - MGET, в последнем случае применяются символы заменители, например, MGET \*.txt (или RFC-18\*.txt, при этом скопируются файлы с RFC-1800.txt до RFC-1899.txt, если таковые существуют в текущем каталоге). Аналогом команды GET в какой-то степени является команда DIR (ls), только она переносит содержимое каталога, что для некоторых операционных систем эквивалентно. При использовании модификации mget проявляйте осторожность - вы можете заблокировать телекоммуникационный канал длительным копированием. Для записи файла в удаленный сервер применяется команда PUT. При операциях обмена обычно используется текущий каталог локальной ЭВМ. В вашем распоряжении всегда имеется возможность поменять местный каталог с помощью команды LCD или ее аналога. Любая команда обмена выполняется в несколько этапов:

1. Формирование канала под управлением клиента, так как именно клиент выдал команду get, dir, put и т.д.
2. Клиент выбирает произвольный номер порта на своем компьютере и осуществляет процедуру passive open для этого порта.
3. Клиент посылает номер порта серверу по каналу управления (порт 21) RFC 959, используя команду PORT. Можно обойтись и без команды PORT (используется тот же порт, что и в командном канале), но это увеличивает задержки и по этой причине не рекомендуется.
4. Сервер получает номер порта по каналу управления и выдает команду active open в указанный порт ЭВМ-клиента. Сервер для канала данных всегда использует порт с номером 20.
5. Уход из FTP производится по команде quit. В приведенном примере файловый обмен не производился, но и команда HELP требует переноса информации (также как и dir), так как вам выдается список команд, доступных на удаленном сервере. Из воспроизведенного списка команд, самая опасная mdelete, так как способна стереть целый каталог. Нетекстовые файлы (архивированные, графические и программные) следует пересылать в режиме binary. Для перевода в этот режим используется одноименная команда. Для перехода из одного каталога в другой на удаленном сервере служит команда cd имя\_каталога, а для возврата в предшествующий cd .. . Например, cd /pub/msdos.
6. Ссылка на объект, доступный через анонимное FTP, обычно записывается в виде:

|  |  |  |
| --- | --- | --- |
| Название ресурса | Имя сервера | Имя каталога в сервере. |

Протокол анонимный FTP описан в RFC 1635.

Некоторые узлы работают как хранилища информации, предназначенной для широкого распространения и доступны без ограничений. Протокол Анонимный FTP (Anonymous FTP) обеспечивает свободный доступ к такой информации. Для этого пользователю достаточно знать лишь имя архивного узла и начало имени файла или путь к каталогу. Архивный узел создает специальную учетную запись с именем пользователя «anonymous» (анонимный), которая позволяет лишь зарегистрироваться, просмотреть ограниченное число каталогов и считать (но не записать) определенное количество файлов. Архивный узел в качестве пароля может принять любую строку, содержащую слово «quest» (гость), либо потребует указать адрес электронной почты.

Для поддержки большего числа разнородных компьютеров FTP определяет несколько файловых структур следующим образом:

• File (файл) — файл передается как сплошной поток байт;

• Record (запись) — файл передается как последовательность записей;

• Page (страница) — файл передается в виде блоков данных; этот режим используется для файлов прямого доступа.

Протокол FTP определяет три возможных режима передачи данных.

• Поточный режим (slream mode) — файл рассматривается как последовательность байт. Этот режим допустимо использовать с любым типом данных. При файловой структуре типа Record применяются специальные двухбайтные управляющие символы, обозначающие конец записи (EOR) и конец файла (EOF). При файловой структуре тина File конец файла обозначается закрытием канала данных передающим компьютером.

• Елочный режим (block mode) — файл передается как набор блоков, каждому из которых предшествует заголовок, содержащий размер блока и дескриптор. Дескриптор определяет, является ли этот блок последним в файле или в записи, а также содержит бит, показывающий надежность данных. Этот бит используется в случае, когда чтение данных большого объема, например определение геологических или атмосферных характеристик, заканчивается ошибкой среды передачи. Дескриптор может также указать, является ли данный блок указателем перезапуска (Rested Marker). Это позволит при необходимости повторной передачи выполнять ее от момента приема последнего указателя перезапуска.

• Сжатый режим (compressed mode) используется для эффективной передачи файлов и оптимального использования сети. Механизм сжатия достаточно прост: строка из *х* повторений одного байта заменяется двумя байтами.

FTP-команды — это ASCII-строки, дополненные необязательными параметрами. Существуют FTP-команды для предоставления «удостоверения личности» пользователя, управления папками, сохранения и запроса файлов, задания типов данных и режима передачи, определения состояния сервера и т. д. Детали можно уточнить в RFC 959, где также указано, что в реализации протокола FTP лишь часть команд — обязательные, остальные же — дополнительные.

Все FTP-команды требуют ответа от сервера, причем некоторые вызывают несколько ответов. Каждый ответ состоит из трехзначного числового кода, после которого следует текстовая строка. Последняя может отображаться для удобства пользователя, а числовой код позволяет управлять конечным автоматом.

Спецификация RFC 2228 описывает набор необязательных FTP-команд, которые могут выполняться FTP-клиентом и FTP-сервером. Эти команды, если они реализованы и клиентом и сервером, обеспечивают аутентификацию (сервер доподлинно будет знать, что клиент именно тот, кем представляется), целостность данных и их конфиденциальность как в канале данных, так и в канале команд (это два ТСР-соединения, используемые протоколом FTP для передачи данных и команд). Таким образом, и сервер и клиент сумеют выявить искажение данных или команд третьей стороной. Данные и команды можно шифровать, так что третья сторона, получив доступ к TCP-пакету, не сможет понять его смысл. Ключ для шифрования пред устанавливается или распределяется по каналу, внешнему по отношению к FTP, например, курьерской или электронной почтой.

Так как разные клиенты обычно пересылают запросы не через одинаковые порты, то невозможно заранее определить, с каким портом FTP-сервер будет инициализировать ТСР-соединение. В этом суть проблемы, так как межсетевые экраны обычно блокируют подключения к неизвестным портам. Спецификация RFC 1379 предлагает модификации FTP-клиентов и серверов, благодаря которым эту проблему можно разрешить.

Протокол TFTP (Trivial File Transfer Protocol) — простой способ передачи файлов между двумя компьютерами. Для управления этим процессом TFTP применяет протокол UDP. Он не содержит механизма идентификации (передачи «удостоверений личности» — имени пользователя и пароля); следовательно, TFTP применяетеся только в областях, доступных любому пользователю. Протокол TFTP позволяет только перемещать файлы и не допускает просмотр списка каталогов, навигацию по ним и т. д.

Протокол TFTP передает данные блоками фиксированной длины — по 512 байт. Он использует окно фиксированного размера, равное одному блоку данных, то есть необходимо подтверждение приема каждого блока, прежде чем отправлять следующий. Пакетами обмениваются оба задействованных компьютера. Один передает блоки данных и принимает подтверждения их успешного приема. Другой принимает блоки данных и передает подтверждении. Каждый компьютер должен выдерживать паузу и повторять передачу, если ожидаемое подтверждение или блок данных не получены. Почти все ошибки останавливают процесс передачи. Если отправка файлов прервалась, то нет возможности ее перезапустить и скопировать оставшуюся часть файла. Спецификация RFC I783 дополняет протокол TFTP механизмом согласования размера блока для передачи файлов. Если этот механизм применяется, то значение размера блока может устанавливаться в пределах от 0 до 65 464 байт. Спецификация RFC 1784 описывает дополнительный механизм согласования длительности пауз — от 1 до 255. Также в RFC 1784 описано, как определить размер передаваемого файла.

Обычно реализации TFTP-сервера предоставляют некоторые средства задания путей, к которым разрешен доступ по протоколу TFTP. Это разумно, учитывая отсутствие защиты для TFTP.

Серверы TFTP «слушают» 69-й порт UDP. Протокол TFTP описан в RFC 1350, а его дополнения, позволяющие расширить возможности TFTP, — в RFC 1705, 1704, 1703 и 1702.

### SMTP

Главной целью протокола simple mail transfer protocol (SMTP, RFC-821, -822) служит надежная и эффективная доставка электронных почтовых сообщений. SMTP является довольно независимой субсистемой и требует только надежного канала связи. Средой для SMTP может служить отдельная локальная сеть, система сетей или весь Интернет.

SMTP базируется на следующей модели коммуникаций: в ответ на запрос пользователя почтовая программа-отправитель устанавливает двухстороннюю связь с программой-приемником (TCP, порт 25). Получателем может быть оконечный или промежуточный адресат. SMTP-команды генерируются отправителем и посылаются получателю. На каждую команду должен быть отправлен и получен отклик.

Когда канал организован, отправитель посылает команду MAIL, идентифицирую себя. Если получатель готов к приему сообщения, он посылает положительное подтверждение. Далее отправитель посылает команду RCPT, идентифицируя получателя почтового сообщения. Если получатель может принять сообщение для оконечного адресата, он выдает снова положительное подтверждение. В противном случае он отвергает получение сообщения для данного адресата, но не вообще почтовой посылки. Взаимодействие с почтовым сервером возможно и в диалоговом режиме.

Почтовое сообщение отправлено без использования доступа к локальной почтовой программе (mail на sun, например). Следует отметить, что работа через порт 25 в данном случае открывает богатые возможности для хакеров. Не имея авторизации, можно выявить клиентов почтового сервера, используя команду VRFY.

SMTP-отправитель и SMTP-получатель могут вести диалог с несколькими оконечными пользователями (рис). Любое почтовое сообщение завершается специальной последовательностью символов. Если получатель успешно завершил прием и обработку почтового сообщения, он посылает положительное подтверждение.

SMTP обеспечивает передачу почтового сообщения непосредственно конечному получателю, когда они соединены непосредственно. В противном случае пересылка может выполняться через одного или более промежуточных "почтовых станций".



Рисунок Схема взаимодействия различных частей почтовой системы

Для решения поставленной задачи SMTP-сервер должен знать имя конечного получателя и название почтового ящика места назначения. Аргументом команды MAIL является адрес отправителя (обратный адрес). Аргументом команды RCPT служит адрес конечного получателя. Обратный адрес используется для посылки сообщения в случае ошибки.

Все отклики имеют цифровые коды. Команды, отклики и имена ЭВМ не чувствительны к тому, строчные или прописные символы использованы при их написании. Это не всегда справедливо при написании имен и адресов получателя.

Многие почтовые системы работают только с ASCII-символами. Если транспортный канал работает с октетами, 7-битные коды будут дополнены нулевым восьмым битом. Именно здесь коренилась проблема пересылки почтовых сообщений на русском языке несколько лет назад(русский алфавит требует 8-битового представления).

Как уже было сказано, процедура отправки почтового сообщения начинается с посылки команды *MAIL*, которая имеет формат:

MAIL <SP> FROM:<reverse-path> <CRLF>,

где <SP> - пробел, <CRLF> - комбинация кодов возврата каретки и перехода на новую строку, а <reverse-path> - обратный путь.

Эта команда сообщает SMTP-получателю, что стартует новая процедура и следует сбросить в исходное состояние все статусные таблицы, буферы и т.д. Если команда прошла, получатель реагирует откликом: 250 OK.

Аргумент может содержать не только адрес почтового ящика, в общем случае является списком адресов ЭВМ-серверов, через которые пришло данное сообщение, включая, разумеется, и адрес почтового ящика отправителя. Первым в списке <reverse-path> стоит адрес ЭВМ-отправителя.

После прохождения команды MAIL посылается команда *RCPT*:

RCPT <SP> TO:<forward-path> <CRLF>

Эта команда указывает адрес конечного получателя (<forward-path>). При благополучном прохождении команды получатель посылает код-отклик 250 OK, и запоминает полученный адрес. Если получатель неизвестен, SMTP-сервер пошлет отклик 550 Failure reply. Команда RCPT может повторяться сколько угодно раз, если адресат не один.

Аргумент может содержать не только адрес почтового ящика, но и маршрутный список ЭВМ по дороге к нему. Первым в этом списке должно стоять имя ЭВМ, получившей данную команду. По завершении этого этапа посылается собственно сообщение:

DATA <CRLF>

При правильном приеме этого сообщения SMTP-сервер реагирует посылкой отклика 354 Intermediate reply (промежуточный отклик), и рассматривает все последующие строки в качестве почтового текста. При получении кода конца текста отправляется отклик: 250 OK.

Признаком конца почтового сообщения является точка в самом начале строки, за которой следует <CRLF>.

В некоторых случаях адрес места назначения может содержать ошибку, но получатель знает правильный адрес. Тогда возможны два варианта отклика:

1. 251 User not local; will forward to <forward-path>

Это означает, что получатель берет на себя ответственность за доставку сообщения. Это случается, когда адресат, например, мигрировал в другую субсеть в пределах зоны действия данного почтового сервера.

2. 551 User not local; please try <forward-path>

Получатель знает правильный адрес и предлагает отправителю переадресовать сообщение по адресу <forward-path>.

SMTP имеет команды для проверки корректности имени адресата (VRFY) и расширения списка адресов (EXPN). Обе команды в качестве аргументов используют строки символов (в некоторых реализациях эти две команды по своей функции идентичны). Для команды VRFY параметром является имя пользователя, а отклик может содержать его полное имя и адрес его почтового ящика.

Основной задачей почты служит доставка сообщений в почтовый ящик адресата. Сходную форму услуги оказывают некоторые ЭВМ, доставляя сообщения на экран терминала (в рамках SMTP). Для посылки сообщений на экран терминала адресата предусмотрено три команды:

1. SEND <SP> FROM:<reverse-path> <CRLF>

Команда SEND требует, чтобы почтовое сообщение было доставлено на терминал. Если терминал адресата не активен в данный момент, то откликом на команду RCPT будет код 450.

2. SOML <SP> FROM:<reverse-path> <CRLF>

Команда Send Or MaiL (SOML) пересылает сообщение на экран адресата, если он активен, в противном случае сообщение будет уложено в его почтовый ящик.

3. SAML <SP> FROM:<reverse-path> <CRLF>

Команда Send And MaiL (SAML) предполагает доставку сообщение на экран терминала адресата и занесение в его почтовый ящик.

QUIT <CRLF>

Выражение может быть маршрутом, имеющим вид "@ONE,@TWO:VANJA@THREE", где ONE, TWO и THREE - имена ЭВМ. Что подчеркивает различие между адресом и маршрутом. Концептуально элементы из <forward-path> переносятся в <reverse-path> при пересылке сообщений от одного SMTP-сервера к другому.

Если SMTP-сервер обнаружит, что доставка сообщения по адресу <forward-path> невозможна, тогда он формирует сообщение о "не доставленном письме", используя <reverse-path>. Следует также помнить, что ни прямой ни обратный адреса-маршруты, вообще говоря, могут не иметь ничего общего с текстом заголовка почтового сообщения.

При определенных условиях и ошибках в задании прямых и обратных адресов-маршрутов возможно зацикливание сообщений об этих ошибках. Чтобы заведомо избежать этого можно выдавать команду MAIL c нулевым обратным маршрутом:

MAIL FROM:< >

Если программа не указали обратного адреса, не следует думать, что это помешает работе почтовой программы и она не будет знать, куда посылать отклики. Ваш обратный IP-адрес указан в каждом пакете, посылаемом адресату!

Некоторые другие команды, используемые в SMTP:

Команда **TURN** используется для того, чтобы поменять местами функции программ, взаимодействовавших по телекоммуникационному каналу. Программа-отправитель становится получателем (после того как она выдаст команду TURN и получит отклик 250), а программа-получатель - отправителем. Если программа не хочет или не может поменять свою функцию, она пошлет отклик 502.

Команда **RESET** (RSET) прерывает текущую процедуру отправки почтового сообщения. Все буферы и таблицы очищаются, получатель должен послать отклик 250 OK.

Команда **HELP** вынуждает получателя послать справочную информацию отправителю команды HELP. Команда может содержать аргумент (имя команды). Команда не изменяет состояния таблиц или буферов.

Команда **NOOP** не оказывает влияния на какие-либо параметры или результаты предшествующих команд, она только вынуждает получателя послать отклик 250 OK. Может использоваться для проверки работоспособности TCP-канала.

Допустимо написание команд строчными или прописными символами, например: MAIL, Mail, mail, MaIl или mAIl.

Для того чтобы программа SMTP-сервера была работоспособна, она должна понимать следующий минимум команд: HELO, MAIL, RCPT, DATA, RSET, NOOP, QUIT.

Предельная длина имени пользователя или домена равна 64 символам. Максимальная длина <reverse-path> или <forward-path> составляет 256 символов, включая разделители (пробелы, точки, запятые и пр.). Командная строка не должна быть длиннее 512 символов. Максимальный размер строки отклика не должен превышать, включая его код и <CRLF>, 512 символов. Максимальная длина строки составляет, включая <CRLF>, 1000 символов. Предельно допустимое число адресатов равно 100, последнее полезно помнить, если вы храните этот список в файле.

# ОСНОВЫ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

## Основные термины и определения

***Автоматизированная система обработки информации (АС)*** - организационно-техническая система, представляющая собой совокупность следующих взаимосвязанных компонентов: технических средств обработки и передачи данных (средств вычислительной техники и связи), методов и алгоритмов обработки в виде соответствующего программного обеспечения, массивов (наборов, баз) данных на различных носителях, персонала и пользователей, объединенных по организационно-структурному, тематическому, технологическому или другим признакам для выполнения автоматизированной обработки данных с целью удовлетворения информационных потребностей государственных органов, общественных или коммерческих организаций (юридических лиц), отдельных граждан (физических лиц) и иных потребителей информации.

***Информация в АС*** - сведения о фактах, событиях, процессах и явлениях в некоторой предметной области, включенные в систему обработки информации, или являющиеся ее результатом в различных формах представления на различных носителях и используемые (необходимые) для оптимизации принимаемых решений в процессе управления объектами данной предметной области.

***Обработка информации в АС*** - совокупность операций (сбор, накопление, хранение, преобразование, отображение, выдача и т.п.), осуществляемых над информацией (сведениями, данными) с использованием средств АС.

***Субъекты информационных отношений*** - государство, государственные органы, государственные, общественные или коммерческие организации (объединения) и предприятия (юридические лица), отдельные граждане (физические лица) и иные субъекты, взаимодействующие с целью совместной обработки информации.

По отношению к информации, обрабатываемой в АС различные субъекты участники информационных отношений могут выступать (возможно одновременно) в качестве:

* Источников информации;
* Пользователей (потребителей) информации;
* Собственников (владельцев, распорядителей) информации;
* Физических и юридических лиц, о которых собирается и обрабатывается информация;
* Владельцев АС и участников процессов обработки и передачи информации и т.д.

***Жизненно важные интересы*** - совокупность потребностей, удовлетворение которых необходимо для надежного обеспечения существования и возможности прогрессивного развития субъекта (личности, организации, общества или государства).

***Доступность информации*** - свойство системы, в которой циркулирует информация (средств и технологии ее обработки), характеризующееся способностью обеспечивать своевременный беспрепятственный доступ к информации субъектов, имеющих на это надлежащие полномочия.

***Целостность информации*** - свойство информации, заключающееся в ее существовании в неискаженном виде (неизменном по отношению к некоторому фиксированному ее состоянию).

***Конфиденциальность информации*** - субъективно определяемая (приписываемая) информации характеристика (свойство), указывающая на необходимость введения ограничений на круг субъектов, имеющих доступ к данной информации, и обеспечиваемая способностью системы (среды) сохранять указанную информацию в тайне от субъектов, не имеющих полномочий на право доступа к ней.

Объективные предпосылки подобного ограничения доступности информации заключены в необходимости защиты законных интересов некоторых субъектов информационных отношений.

***Уязвимость субъекта информационных отношений*** - потенциальная подверженность субъекта нанесению ущерба его жизненно важным интересам посредством воздействия на критичную для него информацию, ее носители и процессы обработки.

***Уязвимость информации*** - подверженность информации воздействию различных дестабилизирующих факторов, которые могут привести к нарушению ее конфиденциальности, целостности, доступности, или неправомерному ее тиражированию.

***Угроза интересам субъектов информационных отношений*** - потенциально возможное событие, действие, процесс или явление, которое посредством воздействия на информацию и другие компоненты АС может привести к нанесению ущерба интересам данных субъектов.

***Угроза безопасности информации***- потенциально возможное событие, действие, процесс или явление, которое может привести к нарушению конфиденциальности, целостности, доступности информации, а также неправомерному ее тиражированию.

***Безопасность субъектов информационных отношений*** - защищенность субъектов информационных отношений от нанесения им материального, морального или иного ущерба путем воздействия на информацию и/или средства ее обработки и передачи.

***Безопасность АС (компьютерной системы)*** - защищенность АС от несанкционированного вмешательства в нормальный процесс ее функционирования, а также от попыток хищения, незаконной модификации или разрушения ее компонентов.

***Безопасность любого компонента (ресурса) АС*** складывается из обеспечения трех его характеристик: конфиденциальности, целостности и доступности.

Конфиденциальность компонента системы заключается в том, что он доступен только тем субъектам доступа (пользователям, программам, процессам), которым предоставлены на то соответствующие полномочия.

Целостность компонента предполагает, что он может быть модифицирован только субъектом, имеющим для этого соответствующие права. Целостность является гарантией корректности (неизменности, работоспособности) компонента в любой момент времени.

Доступность компонента означает, что имеющий соответствующие полномочия субъект может в любое время без особых проблем получить доступ к необходимому компоненту системы (ресурсу).

***Безопасность информационной технологии*** - защищенность технологического процесса переработки информации.

***Безопасность информации*** - защищенность информации от нежелательного (для соответствующих субъектов информационных отношений) ее разглашения (нарушения конфиденциальности), искажения (нарушения целостности), утраты или снижения степени доступности информации, а также незаконного ее тиражирования.

***Доступ к информации*** - ознакомление с информацией (копирование, тиражирование), ее модификация (корректировка) или уничтожение (удаление).

***Правила разграничения доступа*** - совокупность правил, регламентирующих права доступа субъектов к объектам в некоторой системе.

***Разграничение доступа к ресурсам АС*** - это такой порядок использования ресурсов системы, при котором субъекты получают доступ к объектам в строгом соответствии с установленными правилами.

***Объект*** - пассивный компонент системы, единица ресурса автоматизированной системы (устройство, диск, каталог, файл и т.п.), доступ к которому регламентируется правилами разграничения доступа.

***Субъект***- активный компонент системы (пользователь, процесс, программа), действия которого регламентируются правилами разграничения доступа.

***Авторизованный субъект доступа*** - субъект, которому предоставлены соответствующие права доступа к объектам системы (полномочия).

***Доступ к ресурсу*** - получение субъектом доступа возможности манипулировать (использовать, управлять, изменять характеристики и т.п.) Данным ресурсом.

***Несанкционированный доступ (НСД)***- доступ субъекта к объекту в нарушение установленных в системе правил разграничения доступа.

***Несанкционированное действие*** - действие субъекта в нарушение установленных в системе правил обработки информации.

***Нарушитель***- это лицо (субъект), которое предприняло (пыталось предпринять) попытку несанкционированного доступа к ресурсам системы (попытку выполнения запрещенных ему действий с данным ресурсом) по ошибке, незнанию или осознанно со злым умыслом (из корыстных интересов) или без такового (ради игры или с целью самоутверждения и т.п.) И использовавшее для этого различные возможности, методы и средства (чисто агентурные методы получения сведений, технические средства перехвата без модификации компонентов системы, штатные средства и недостатки систем защиты, подключение к каналам передачи данных, внедрение программных закладок и использование специальных инструментальных и технологических программ и т.п.).

***Злоумышленник*** - нарушитель, действующий умышленно из корыстных побуждений.

***Система защиты АС (информации)*** - совокупность (комплекс) специальных мер правового (законодательного) и административного характера, организационных мероприятий, физических и технических (программных и аппаратных) средств защиты, а также специального персонала, предназначенных для обеспечения безопасности АС (циркулирующей в АС информации).

***Защита информации*** - непрерывный процесс построения, поддержки нормального функционирования и совершенствования системы защиты информации.

***Цель защиты АС (информации)*** - предотвращение или минимизация наносимого ущерба (прямого или косвенного, материального, морального или иного) субъектам информационных отношений посредством нежелательного воздействия на компоненты АС, а также разглашения (утечки), искажения (модификации), утраты (снижения степени доступности) или незаконного тиражирования информации.

***Правовые меры защиты информации*** - действующие в стране законы, указы и другие нормативные акты, регламентирующие правила обращения с информацией и ответственность за их нарушения, препятствующие тем самым неправомерному ее использованию и являющиеся сдерживающим фактором для потенциальных нарушителей.

***Морально-этические меры защиты информации*** - традиционно сложившиеся в стране или обществе нормы поведения и правила обращения с информацией. Эти нормы большей частью не являются обязательными, как законодательно утвержденные нормы, однако, их несоблюдение ведет обычно к падению авторитета, престижа человека, группы лиц или организации. Морально-этические нормы бывают как неписаные (например, общепризнанные нормы честности, патриотизма и т.п.), так и писаные, то есть оформленные в некоторый свод (устав) правил или предписаний.

***Организационные (административные) меры защиты*** - это меры, регламентирующие процессы функционирования системы обработки данных, использование ее ресурсов, деятельность персонала, а также порядок взаимодействия пользователей с системой таким образом, чтобы в наибольшей степени затруднить или исключить возможность реализации угроз безопасности циркулирующей в ней информации.

***Физические меры защиты*** - это разного рода механические, электро- или электронно-механические устройства и сооружения, специально предназначенные для создания физических препятствий на возможных путях проникновения и доступа потенциальных нарушителей к компонентам АС и защищаемой информации, а также технические средства визуального наблюдения, связи и охранной сигнализации.

***Технические (аппаратно-программные) средства защиты*** - различные электронные устройства и специальные программы, входящие в состав АС, которые выполняют (самостоятельно или в комплексе с другими средствами) функции защиты информации (идентификацию и аутентификацию пользователей, разграничение доступа к ресурсам, регистрацию событий, криптографическое закрытие информации и т.д.).

***Администратор безопасности*** - лицо или группа лиц, ответственных за обеспечение безопасности системы, за реализацию и непрерывность соблюдения установленных административных мер защиты и осуществляющих постоянную организационную поддержку функционирования применяемых физических и технических средств защиты.

## Особенности современных АС как объекта защиты

Как показывает анализ, большинство современных автоматизированных систем обработки информации в общем случае представляет собой территориально распределенные системы интенсивно взаимодействующих (синхронизирующихся) между собой по данным (ресурсам) и управлению (событиям) локальных вычислительных сетей (ЛВС) и отдельных ЭВМ.

В распределенных АС возможны все "традиционные" для локально расположенных (централизованных) вычислительных систем способы несанкционированного вмешательства в их работу и доступа к информации. Кроме того, для них характерны и новые специфические каналы проникновения в систему и несанкционированного доступа к информации, наличие которых объясняется целым рядом их особенностей.

Перечислим основные из особенностей распределенных АС:

* территориальная разнесенность компонентов системы и наличие интенсивного обмена информацией между ними;
* широкий спектр используемых способов представления, хранения и передачи информации;
* интеграция данных различного назначения, принадлежащих различным субъектам, в рамках единых баз данных и, наоборот, размещение необходимых некоторым субъектам данных в различных удаленных узлах сети;
* абстрагирование владельцев данных от физических структур и места размещения данных;
* использование режимов распределенной обработки данных;
* участие в процессе автоматизированной обработки информации большого количества пользователей и персонала различных категорий;
* непосредственный и одновременный доступ к ресурсам (в том числе и информационным) большого числа пользователей (субъектов) различных категорий;
* высокая степень разнородности используемых средств вычислительной техники и связи, а также их программного обеспечения;
* отсутствие специальной аппаратной поддержки средств защиты в большинстве типов технических средств, широко используемых в АС.

В общем случае АС состоят из следующих основных структурно-функциональных элементов:

* рабочих станций - отдельных ЭВМ или удаленных терминалов сети, на которых реализуются автоматизированные рабочие места пользователей (абонентов, операторов);
* серверов или Host машин (служб файлов, печати, баз данных и т.п.) не выделенных (или выделенных, то есть не совмещенных с рабочими станциями) высокопроизводительных ЭВМ, предназначенных для реализации функций хранения, печати данных, обслуживания рабочих станций сети и т.п. действий;
* межсетевых мостов (шлюзов, центров коммутации пакетов, коммуникационных ЭВМ) - элементов, обеспечивающих соединение нескольких сетей передачи данных, либо нескольких сегментов одной и той же сети, имеющих различные протоколы взаимодействия;
* каналов связи (локальных, телефонных, с узлами коммутации и т.д.).

Рабочие станции являются наиболее доступными компонентами сетей и именно с них могут быть предприняты наиболее многочисленные попытки совершения несанкционированных действий. С рабочих станций осуществляется управление процессами обработки информации, запуск программ, ввод и корректировка данных, на дисках рабочих станций могут размещаться важные данные и программы обработки. На видеомониторы и печатающие устройства рабочих станций выводится информация при работе пользователей (операторов), выполняющих различные функции и имеющих разные полномочия по доступу к данным и другим ресурсам системы. Именно поэтому рабочие станции должны быть надежно защищены от доступа посторонних лиц и содержать средства разграничения доступа к ресурсам со стороны законных пользователей, имеющих разные полномочия. Кроме того, средства защиты должны предотвращать нарушения нормальной настройки рабочих станций и режимов их функционирования, вызванные неумышленным вмешательством неопытных (невнимательных) пользователей.

В особой защите нуждаются такие привлекательные для злоумышленников элементы сетей как серверы (Host - машины) и мосты. Первые - как концентраторы больших объемов информации, вторые - как элементы, в которых осуществляется преобразование (возможно через открытую, нешифрованную форму представления) данных при согласовании протоколов обмена в различных участках сети.

Непосредственные случайные воздействия персонала и преднамеренные воздействия злоумышленников на выделенные серверы и мосты можно считать маловероятными. В то же время, надо ожидать массированной атаки на серверы и мосты с использованием средств удаленного доступа. Здесь злоумышленники прежде всего могут искать возможности повлиять на работу различных подсистем серверов и мостов, используя недостатки протоколов обмена и средств разграничения удаленного доступа к ресурсам и системным таблицам. Использоваться могут все возможности и средства, от стандартных (без модификации компонентов) до подключения специальных аппаратных средств (каналы, как правило, слабо защищены от подключения) и применения высококлассных программ для преодоления системы защиты.

Конечно, сказанное выше не означает, что не будет попыток внедрения аппаратных и программных закладок в сами мосты и серверы, открывающих дополнительные широкие возможности по несанкционированному удаленному доступу. Закладки могут быть внедрены как с удаленных станций (посредством вирусов или иным способом), так и непосредственно в аппаратуру и программы серверов при их ремонте, обслуживании, модернизации, переходе на новые версии программного обеспечения, смене оборудования.

Каналы и средства связи также нуждаются в защите. В силу большой пространственной протяженности линий связи (через неконтролируемую или слабо контролируемую территорию) практически всегда существует возможность подключения к ним, либо вмешательства в процесс передачи данных.

## Угрозы безопасности информации

Под угрозой (вообще) обычно понимают потенциально возможное событие, действие (воздействие), процесс или явление, которое может привести к нанесению ущерба чьим-либо интересам.

Угрозой интересам субъектов информационных отношений будем называть потенциально возможное событие, процесс или явление, которое посредством воздействия на информацию или другие компоненты АС может прямо или косвенно привести к нанесению ущерба интересам данных субъектов.

В силу особенностей современных АС, перечисленных выше, существует значительное число различных видов угроз безопасности субъектов информационных отношений.

Нарушением безопасности (или просто нарушением) будем называть реализацию угрозы безопасности.

Основными видами угроз безопасности АС и информации (угроз интересам субъектов информационных отношений) являются:

* стихийные бедствия и аварии (наводнение, ураган, землетрясение, пожар и т.п.);
* сбои и отказы оборудования (технических средств) АС;
* последствия ошибок проектирования и разработки компонентов АС (аппаратных средств, технологии обработки информации, программ, структур данных и т.п.);
* ошибки эксплуатации (пользователей, операторов и другого персонала);
* преднамеренные действия нарушителей и злоумышленников (обиженных лиц из числа персонала, преступников, шпионов, диверсантов и т.п.).

Все множество потенциальных угроз по природе их возникновения разделяется на два класса: естественные (объективные) и искусственные (субъективные).

Естественные угрозы - это угрозы, вызванные воздействиями на АС и ее элементы объективных физических процессов или стихийных природных явлений, независящих от человека.

Искусственные угрозы - это угрозы АС, вызванные деятельностью человека. Среди них, исходя из мотивации действий, можно выделить:

* **непреднамеренные** (неумышленные, случайные) угрозы, вызванные ошибками в проектировании АС и ее элементов, ошибками в программном обеспечении, ошибками в действиях персонала и т.п.;
* **преднамеренные** (умышленные) угрозы, связанные с корыстными устремлениями людей (злоумышленников).

Источники угроз по отношению к АС могут быть **внешними** или **внутренними** (компоненты самой АС - ее аппаратура, программы, персонал).

Основные непреднамеренные искусственные угрозы АС (действия, совершаемые людьми случайно, по незнанию, невнимательности или халатности, из любопытства, но без злого умысла):

неумышленные действия, приводящие к частичному или полному отказу системы или разрушению аппаратных, программных, информационных ресурсов системы (неумышленная порча оборудования, удаление, искажение файлов с важной информацией или программ, в том числе системных и т.п.);

неправомерное отключение оборудования или изменение режимов работы устройств и программ;

1. неумышленная порча носителей информации;
2. запуск технологических программ, способных при некомпетентном использовании вызывать потерю работоспособности системы (зависания или зацикливания) или осуществляющих необратимые изменения в системе (форматирование или реструктуризацию носителей информации, удаление данных и т.п.);
3. нелегальное внедрение и использование неучтенных программ (игровых, обучающих, технологических и др., не являющихся необходимыми для выполнения нарушителем своих служебных обязанностей) с последующим необоснованным расходованием ресурсов (загрузка процессора, захват оперативной памяти и памяти на внешних носителях);
4. заражение компьютера вирусами;
5. неосторожные действия, приводящие к разглашению конфиденциальной информации, или делающие ее общедоступной;
6. разглашение, передача или утрата атрибутов разграничения доступа (паролей, ключей шифрования, идентификационных карточек, пропусков и т.п.);
7. проектирование архитектуры системы, технологии обработки данных, разработка прикладных программ, с возможностями, представляющими опасность для работоспособности системы и безопасности информации;
8. игнорирование организационных ограничений (установленных правил) при работе в системе;
9. вход в систему в обход средств защиты (загрузка посторонней операционной системы со сменных магнитных носителей и т.п.);
10. некомпетентное использование, настройка или неправомерное отключение средств защиты персоналом службы безопасности;
11. пересылка данных по ошибочному адресу абонента (устройства);
12. ввод ошибочных данных;
13. неумышленное повреждение каналов связи.

Основные возможные пути умышленной дезорганизации работы, вывода системы из строя, проникновения в систему и несанкционированного доступа к информации:

1. физическое разрушение системы (путем взрыва, поджога и т.п.) или вывод из строя всех или отдельных наиболее важных компонентов компьютерной системы (устройств, носителей важной системной информации, лиц из числа персонала и т.п.);
2. отключение или вывод из строя подсистем обеспечения функционирования вычислительных систем (электропитания, охлаждения и вентиляции, линий связи и т.п.);
3. действия по дезорганизации функционирования системы (изменение режимов работы устройств или программ, забастовка, саботаж персонала, постановка мощных активных радиопомех на частотах работы устройств системы и т.п.);
4. внедрение агентов в число персонала системы (в том числе, возможно, и в административную группу, отвечающую за безопасность);
5. вербовка (путем подкупа, шантажа и т.п.) персонала или отдельных пользователей, имеющих определенные полномочия;
6. применение подслушивающих устройств, дистанционная фото- и видеосъемка и т.п.;
7. перехват побочных электромагнитных, акустических и других излучений устройств и линий связи, а также наводок активных излучений на вспомогательные технические средства, непосредственно не участвующие в обработке информации (телефонные линии, сети питания, отопления и т.п.);
8. перехват данных, передаваемых по каналам связи, и их анализ с целью выяснения протоколов обмена, правил вхождения в связь и авторизации пользователя и последующих попыток их имитации для проникновения в систему;
9. хищение носителей информации (магнитных дисков, лент, микросхем памяти, запоминающих устройств и целых ПЭВМ);
10. несанкционированное копирование носителей информации;
11. хищение производственных отходов (распечаток, записей, списанных носителей информации и т.п.);
12. чтение остаточной информации из оперативной памяти и с внешних запоминающих устройств;
13. чтение информации из областей оперативной памяти, используемых операционной системой (в том числе подсистемой защиты) или другими пользователями, в асинхронном режиме используя недостатки мультизадачных операционных систем и систем программирования;
14. незаконное получение паролей и других реквизитов разграничения доступа (агентурным путем, используя халатность пользователей, путем подбора, путем имитации интерфейса системы и т.д.) с последующей маскировкой под зарегистрированного пользователя ("маскарад");
15. несанкционированное использование терминалов пользователей, имеющих уникальные физические характеристики, такие как номер рабочей станции в сети, физический адрес, адрес в системе связи, аппаратный блок кодирования и т.п.;
16. вскрытие шифров криптозащиты информации;
17. внедрение аппаратных спецвложений, программных "закладок" и "вирусов" ("троянских коней" и "жучков"), то есть таких участков программ, которые не нужны для осуществления заявленных функций, но позволяющих преодолевать систему защиты, скрытно и незаконно осуществлять доступ к системным ресурсам с целью регистрации и передачи критической информации или дезорганизации функционирования системы;
18. незаконное подключение к линиям связи с целью работы "между строк", с использованием пауз в действиях законного пользователя от его имени с последующим вводом ложных сообщений или модификацией передаваемых сообщений;
19. незаконное подключение к линиям связи с целью прямой подмены законного пользователя путем его физического отключения после входа в систему и успешной аутентификации с последующим вводом дезинформации и навязыванием ложных сообщений.

Следует заметить, что чаще всего для достижения поставленной цели злоумышленник использует не один, а некоторую совокупность из перечисленных выше путей.

Все каналы проникновения в систему и утечки информации разделяют на прямые и косвенные. Под косвенными понимают такие каналы, использование которых не требует проникновения в помещения, где расположены компоненты системы. Для использования прямых каналов такое проникновение необходимо. Прямые каналы могут использоваться без внесения изменений в компоненты системы или с изменениями компонентов.

По типу основного средства, используемого для реализации угрозы все возможные каналы можно условно разделить на три группы, где таковыми средствами являются: человек, программа или аппаратура.

Классификация видов нарушений работоспособности систем и несанкционированного доступа к информации по объектам воздействия и способам нанесения ущерба безопасности приведена в таблице 2.

По способу получения информации потенциальные каналы доступа можно разделить на:

* физический;
* электромагнитный (перехват излучений);
* информационный (программно-математический).

При контактном НСД (физическом, программно-математическом) возможные угрозы информации реализуются путем доступа к элементам АС, к носителям информации, к самой вводимой и выводимой информации (и результатам), к программному обеспечению (в том числе к операционным системам), а также путем подключения к линиям связи.

При бесконтактном доступе (например, по электромагнитному каналу) возможные угрозы информации реализуются перехватом излучений аппаратуры АС, в том числе наводимых в токопроводящих коммуникациях и цепях питания, перехватом информации в линиях связи, вводом в линии связи ложной информации, визуальным наблюдением (фотографированием) устройств отображения информации, прослушиванием переговоров персонала АС и пользователей.

Преступления, в том числе и компьютерные, совершаются людьми. Пользователи системы и ее персонал, с одной стороны, являются составной частью, необходимым элементом АС. С другой стороны, они же являются основной причиной и движущей силой нарушений и преступлений. В этом смысле вопросы безопасности автоматизированных систем есть суть вопросы человеческих отношений и человеческого поведения.

## Неформальная модель нарушителя в АС

 Нарушитель - это лицо, предпринявшее попытку выполнения запрещенных операций (действий) по ошибке, незнанию или осознанно со злым умыслом (из корыстных интересов) или без такового (ради игры или удовольствия, с целью самоутверждения и т.п.) и использующее для этого различные возможности, методы и средства.

Злоумышленником будем называть нарушителя, намеренно идущего на нарушение из корыстных побуждений.

Неформальная модель нарушителя отражает его практические и теоретические возможности, априорные знания, время и место действия и т.п. Для достижения своих целей нарушитель должен приложить некоторые усилия, затратить определенные ресурсы. Исследовав причины нарушений, можно либо повлиять на сами эти причины (конечно если это возможно), либо точнее определить требования к системе защиты от данного вида нарушений или преступлений.

В каждом конкретном случае, исходя из конкретной технологии обработки информации, может быть определена модель нарушителя, которая должна быть адекватна реальному нарушителю для данной АС.

При разработке модели нарушителя определяются:

* предположения о категориях лиц, к которым может принадлежать нарушитель;
* предположения о мотивах действий нарушителя (преследуемых нарушителем целях);
* предположения о квалификации нарушителя и его технической оснащенности (об используемых для совершения нарушения методах и средствах);
* ограничения и предположения о характере возможных действий нарушителей.
* По отношению к АС нарушители могут быть внутренними (из числа персонала системы) или внешними (посторонними лицами).

Можно выделить три основных мотива нарушений: безответственность, самоутверждение и корыстный интерес.

При нарушениях, вызванных безответственностью, пользователь целенаправленно или случайно производит какие-либо разрушающие действия, не связанные тем не менее со злым умыслом. В большинстве случаев это следствие некомпетентности или небрежности.

Некоторые пользователи считают получение доступа к системным наборам данных крупным успехом, затевая своего рода игру "пользователь - против системы" ради самоутверждения либо в собственных глазах, либо в глазах коллег.

Нарушение безопасности АС может быть вызвано и корыстным интересом пользователя системы. В этом случае он будет целенаправленно пытаться преодолеть систему защиты для доступа к хранимой, передаваемой и обрабатываемой в АС информации. Даже если АС имеет средства, делающие такое проникновение чрезвычайно сложным, полностью защитить ее от проникновения практически невозможно.

Всех нарушителей можно классифицировать следующим образом.

По уровню знаний об АС :

* знает функциональные особенности АС, основные закономерности формирования в ней массивов данных и потоков запросов к ним, умеет пользоваться штатными средствами;
* обладает высоким уровнем знаний и опытом работы с техническими средствами системы и их обслуживания;
* обладает высоким уровнем знаний в области программирования и вычислительной техники, проектирования и эксплуатации автоматизированных информационных систем;
* знает структуру, функции и механизм действия средств защиты, их сильные и слабые стороны.

По уровню возможностей (используемым методам и средствам):

* применяющий чисто агентурные методы получения сведений;
* применяющий пассивные средства (технические средства перехвата без модификации компонентов системы);
* использующий только штатные средства и недостатки систем защиты для ее преодоления (несанкционированные действия с использованием разрешенных средств), а также компактные магнитные носители информации, которые могут быть скрытно пронесены через посты охраны;
* применяющий методы и средства активного воздействия (модификация и подключение дополнительных технических средств, подключение к каналам передачи данных, внедрение программных закладок и использование специальных инструментальных и технологических программ).

По времени действия:

* в процессе функционирования АС (во время работы компонентов системы);
* в период неактивности компонентов системы (в нерабочее время, во время плановых перерывов в ее работе, перерывов для обслуживания и ремонта и т.п.);
* как в процессе функционирования АС, так и в период неактивности компонентов системы.

По месту действия:

* без доступа на контролируемую территорию организации;
* с контролируемой территории без доступа в здания и сооружения;
* внутри помещений, но без доступа к техническим средствам АС;
* с рабочих мест конечных пользователей (операторов) АС;
* с доступом в зону данных (баз данных, архивов и т.п.);
* с доступом в зону управления средствами обеспечения безопасности АС.

Могут учитываться следующие ограничения и предположения о характере действий возможных нарушителей:

* работа по подбору кадров и специальные мероприятия затрудняют возможность создания коалиций нарушителей, т.е. объединения (сговора) и целенаправленных действий по преодолению подсистемы защиты двух и более нарушителей;
* нарушитель, планируя попытки НСД, скрывает свои несанкционированные действия от других сотрудников;
* НСД может быть следствием ошибок пользователей, администраторов, эксплуатирующего и обслуживающего персонала, а также недостатков принятой технологии обработки информации и т.д.

Определение конкретных значений характеристик возможных нарушителей в значительной степени субъективно. Модель нарушителя, построенная с учетом особенностей конкретной предметной области и технологии обработки информации, может быть представлена перечислением нескольких вариантов его облика. Каждый вид нарушителя должен быть охарактеризован значениями характеристик, приведенных выше.

# АНАЛИЗ УЯЗВИМОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ И ОЦЕНКА РИСКОВ

## Классификация сетевых атак

Снифферы пакетов

Сниффер пакетов представляет собой прикладную программу, которая использует сетевую карту, работающую в режиме promiscuous mode (в этом режиме все пакеты, полученные по физическим каналам, сетевой адаптер отправляет приложению для обработки). При этом сниффер перехватывает все сетевые пакеты, которые передаются через определенный домен. В настоящее время снифферы работают в сетях на вполне законном основании. Они используются для диагностики неисправностей и анализа трафика. Однако ввиду того, что некоторые сетевые приложения передают данные в текстовом формате (telnet, FTP, SMTP, POP3 и т. д.) с помощью сниффера можно узнать полезную, а иногда и конфиденциальную информацию (например, имена пользователей и пароли).

Смягчить угрозу сниффинга пакетов можно с помощью таких средств, как:

• Аутентификация — это сильные средства защиты от сниффинга пакетов, поскольку их трудно игнорировать или обойти. Примером такой аутентификации являются однократные пароли (ОТР - One-Time Passwords). OTP - это технология двухфакторной аутентификации, основанной на сочетании того, что у вас есть, с тем, что вы знаете. Типичным примером двухфакторной аутентификации является работа обычного банкомата, который опознает вас, во-первых, по вашей пластиковой карточке и, во-вторых, по вводимому вами ПИН-коду.

• Коммутируемая инфраструктура — Это также способ борьбы со сниффингом пакетов в сетевой среде. При этом острота угрозы заметно снижается. Ведь, к примеру, во всей организации используется коммутируемый Ethernet, то хакеры получают тогда доступ лишь к трафику, поступающему на тот порт, к которому они подключены. И не далее.

• Анти-снифферы — Способ борьбы со сниффингом, заключающийся в установке аппаратных или программных средств, распознающих снифферы, работающие в вашей сети. Эти средства не могут полностью ликвидировать угрозу, но, как и многие другие средства сетевой безопасности, они включаются в общую систему защиты. Так называемые «анти-снифферы» измеряют время реагирования хостов и определяют, не приходится ли хостам обрабатывать «лишний» трафик.

• Криптография - Это самый эффективный способ борьбы со сниффингом пакетов, который правда, не предотвращает перехвата и не распознает работу снифферов, но делает ее бесполезной. Если канал связи является криптографически защищен, то это значит, что хакер перехватывает не сообщение, а зашифрованный текст.

IР-спуфинг

IP-спуфинг происходит, когда хакер, находящийся внутри корпорации или вне ее выдает себя за санкционированного пользователя. Это можно сделать двумя способами. Во-первых, хакер может воспользоваться IP-адресом, находящимся в пределах диапазона санкционированных IP-адресов, или авторизованным внешним адресом, которому разрешается доступ к определенным сетевым ресурсам. Атаки IР-спуфинга часто становятся отправными для прочих атак.

Обычно IP-спуфинг ограничивается вставкой ложной информации или вредоносных команд в обычный поток данных, передаваемых между клиентским и серверным приложением или по каналу связи между одноранговыми устройствами. Для двусторонней связи хакер должен изменить все таблицы маршрутизации, чтобы направить трафик на ложный IP-адрес.

Угрозу спуфинга можно ослабить (но не устранить) с помощью следующих мер:

• Контроль доступа - самый простой способ предотвращения IP-спуфинга, состоящий в правильной настройке управления доступом. Чтобы снизить эффективность IP-спуфинга, надо настроить контроль доступа на отсечение любого трафика, поступающего из внешней сети с таким исходным адресом, который по идее должен располагаться внутри вашей сети.

• Фильтрация RFC 2827 - способ пресечь попытки спуфинга чужих сетей пользователями вашей. Для этого необходимо только отбраковывать любой исходящий трафик, адрес которого не является одним из IP-адресов вашей организации. Этот тип фильтрации, и он известен под названием «RFC 2827», может выполнять и провайдер.

• Введение дополнительных методов аутентификации - наиболее эффективный метод борьбы с IP-спуфингом тот же, что и в случае со сниффингом пакетов: необходимо сделать атаку абсолютно неэффективной. IP-спуфинг может функционировать только при условии, что аутентификация происходи т на базе IP-адресов. Поэтому внедрение дополнительных методов аутентификации делает этот вид атак бесполезными. Лучшим видом дополнительной аутентификации является криптографическая. Если она невозможна, хорошие результаты может дать двухфакторная аутентификация с использованием одноразовых паролей.

Отказ в обслуживании (Denial of Service — DoS)

Для организации DoS требуется минимум знаний и умений. Тем не менее именно простота их реализации и причиняемый ими огромный вред привлекают к DoS пристальное внимание администраторов, отвечающих за сетевую безопасность. Наиболее известные разновидности DoS:

• TCP SYN Flood - имеет место, в случае если клиент пытается установить TCP-соединение с сервером, что требует обмена определенной последовательностью сообщений. Сначала клиентская система посылает SYN-пакет на сервер. После этого сервер подтверждает получение SYN-пакета, отсылая SYN-ACK-сообщение клиенту. Затем клиент завершает установку соединения, отвечая сообщением ACK, и затем снова должен произойти обмен данными. В точке, где система сервера послала подтверждение (SYN-ACK) назад клиенту, но еще не получила сообщения ACK, устанавливается полуоткрытое соединение. «Фишка» в том, что параметры, касающиеся всех ждущих обработки соединений, располагаются в оперативной памяти сервера, которая не безразмерна, разумеется. Если преднамеренно создать большое количество частично открытых соединений, то память переполнится, и система подвиснет.

• Tribe Flood Network (TFN) и Tribe Flood Network 2000 (TFN2K) Использование TFN-атаки дает возможность генерировать пакеты с фальшивыми IP-адресами источника. Механизм атаки приблизительно таков: злонамеренный пользователь посылает с главного компьютера команды нападения на список TFN-серверов или демонов. Затем демоны генерируют указанный тип DoS-атаки на один или несколько IP-адресов жертв. IP-адреса и порты источника атаки могут изменяться совершенно случайным образом, как и размеры пакетов,

• Stacheldracht объединяет в себе особенности некоторых DoS-атак, в том числе TFN, шифрование связи между нападающим и главными серверами Stacheldraht и автоматическое обновление агентов. Начальный этап атаки включает активное массированное проникновение в большое количество систем для последующего использования их при атаке. Затем следует заключительный этап, в ходе которого «порабощенные» системы используются для атаки на один или несколько объектов.

Атаки DoS отличаются от других атак. Они не нацелены на осуществление доступа к сети или на получение из нее какой-либо информации. Атака DoS делает сеть недоступной для обычного ее использования из -за наступающего вследствии подобной агрессии превышения допустимых параметров функционирования сети, ее операционной системы или приложения. Эти атаки трудно предотвратить, поскольку необходима координация действий с провайдером. Если трафик, переполняющий вашу сеть, не перекрыть у провайдера, то на входе в сеть этого уже не сделаешь, потому что полоса пропускания будет занята полностью. Когда атака подобного типа проводится одновременно через множество устройств, то она квалифицируется специалистами как «распределенная» (DDoS - distributed DoS).

Угрозу атак типа DoS можно снижать тремя следующими способами:

• Использование функции анти-спуфинга — правильная конфигурация функций анти-спуфинга на маршрутизаторах и межсетевых экранах предполагает включение, как минимум, фильтрации RFC 2827. В этом случае хакер уже не сможет замаскировать свою истинную личность и вряд ли решится провести атаку.

• Реализация функции анти-DoS - правильная конфигурация функций анти-DoS на маршрутизаторах и межсетевых экранах может ограничить эффективность атак. Эти функции часто уменьшают число полуоткрытых каналов в любой момент времени.

• Ограничение объема трафика (traffic rate limiting) — организация может попросить провайдера (ISP) ограничить объем трафика. Этот тип фильтрации позволяет уменьшить объем некритического трафика, в сети. Распространенное ограничение объемов трафика ICMP, который используется только для диагностических целей.

Парольные атаки

Хакеры могут проводить парольные атаки с помощью целого ряда методов. Таких как простой перебор (brute force attack), троянский конь, IP-спуфинг и сниффинг пакетов. Часто для такой атаки используется специальная программа, с помощью которой стремится получить доступ к ресурсу общего пользования. Если это удается, то хакер становится как бы обычным пользователем, пароль которого был законно подобран. Если этот пользователь имеет значительные привилегии доступа, то хакер может «застолбить» для себя доступ и на будущее, который будет действовать даже если пользователь изменит свой па роль и логин

Парольных атак можно избежать вообще, если не пользоваться паролями в текстовой форме. Одноразовые пароли и/или криптографическая аутентификация могут практически свести на нет угрозу таких атак. Минимальная длина пароля должна быть не менее восьми символов. Пароль должен иметь символы верхнего регистра, цифры и специальные символы (#, %, $ и т. д.). С точки зрения администратора, возмложно использование средства LOphtCrack, которое часто применяют хакеры для подбора паролей в среде Windows NT. Это средство подскажет, легко ли подобрать пароль, выбранный пользователем.

Атаки типа Man-in-the-Middle

Для атаки типа Man-in-the-Middle хакеру нужен доступ к пакетам, передаваемым по сети. Такой доступ ко всем пакетам, передаваемым от провайдера в любую другую сеть. Для атак этого типа часто используются снифферы пакетов, транспортные протоколы и протоколы маршрутизации. Атаки проводятся с целью кражи информации, перехвата текущей сессии и получения доступа к частным сетевым ресурсам, а также для анализа трафика и получения информации о сети и ее пользователях, для проведения атак типа DoS, искажения передаваемых данных и ввода несанкционированной информации в сетевые сессии.

Эффективно бороться с атаками типа Man-in-the-Middle можно только с помощью криптографии. Если хакер перехватит данные зашифрованной сессии, у него на экране появится лишь бессмысленный набор символов. Следует иметь в виду однако, что если хакер получит информацию о криптографической сессии (например, ключ сессии), то это может сделать возможной атаку Man-in-the-Middle даже в зашифрованной среде.

Атаки на уровне приложений

Атаки на уровне приложений могут проводиться несколькими способами. Самый распространенный из них состоит в использовании хорошо известных слабостей серверного программного обеспечения (sendmail, HTTP, FTP). Используя эти слабости, хакеры могут получить доступ к компьютеру от имени пользователя, работающего с приложением (обычно э то бывает не простой пользователь, а привилегированный администратор с правами системного доступа). Сведения об атаках на уровне приложений широко публикуются, чтобы дать возможность администраторам устранить проблему с помощью коррекционных модулей (патчей). К сожалению, многие хакеры также имеют доступ к этим сведениям, что позволяет им обучаться. Самое главное здесь - хорошее системное администрирование. Вот некоторые меры, которые можно предпринять, чтобы снизить уязвимость для атак подобного типа:

• просмотр лог-файлов операционных систем и сетевых логфайлов и их анализ с помощью специальных аналитических приложений;

• использование самых свежих версий операционных систем и приложений и самых последних коррекционных модулей (патч);

• использование кроме системного администрирования, систем распознавания атак (IDS). Существуют две взаимно дополняющие друг друга технологии IDS:

• сетевая система IDS (NIDS) отслеживает все пакеты, про-ходящие через определенный домен. Когда система NIDS видит пакет или серию пакетов, совпадающих с сигнатурой известной или вероятной атаки, она генерирует сигнал тревоги и/или прекращает сессию;

• хост-система IDS (HIDS) защищает хост с помощью программных агентов. Эта система борется только с атаками против одного хоста;

Сетевая разведка

Сетевой разведкой у хакеров называется сбор информации о сети в виде соответствующих общедоступных данных и приложений. При подготовке атаки против какой-либо сети, как правило, хакер пытается получить о ней информации как можно больше. Сетевая разведка проводится в форме запросов DNS, эхо-тестирования (ping sweep) и сканирования портов. Запросы DNS помогают понять, кто владеет тем или иным доменом и какие адреса этому домену присвоены. Эхо-тестирование (ping sweep) адресов, раскрытых с помощью DNS, позволяет увидеть, какие хосты реально работают в данной среде. Получив список хостов, хакер использует средства сканирования портов, чтобы составить полный список услуг, поддерживаемых этими хостами. И, наконец, хакер анализирует характеристики приложений, работающих на хостах. В результате добывается информация, которую можно использовать для взлома.

Полностью избавиться от сетевой разведки невозможно. Однако возможно использование систем IDS на уровне сети и хостов, которые обычно хорошо справляются с задачей уведомления администратора о ведущейся сетевой разведке, что позволяет лучше подготовиться к предстоящей атаке и оповестить провайдера (ISP), в сети которого установлена система, проявляющая чрезмерное любопытство.

Злоупотребление доверием

Это злонамеренное использование отношений доверия, существующих в сети. Классическим примером является ситуация в периферийной части корпоративной сети. В этом сегменте часто располагаются серверы DNS, SMTP и HTTP. Поскольку все они принадлежат к одному и тому же сегменту, взлом одного из них приводит к взлому и всех остальных, так как эти серверы доверяют другим системам своей сети. Риск злоупотребления доверием можно снизить за счет более жесткого контроля уровней доверия в пределах своей сети. Системы, расположенные с внешней стороны межсетевого экрана, никогда не должны пользоваться абсолютным доверием со стороны защищенных экраном систем. Отношения доверия должны ограничиваться определенными протоколами и, по возможности, аутентифицироваться не только по IP-адресам, но и по другим параметрам.

Переадресация портов

Переадресация портов представляет собой разновидность злоупотребления доверием, когда взломанный хост используется для передачи через межсетевой экран трафика, который в противном случае был бы обязательно отбракован. Основным способом борьбы с переадресацией портов является использование надежных моделей доверия. Кроме того, помешать хакеру установить на хосте свои программные средства может хост-система IDS (HIDS).

Несанкционированный доступ

Несанкционированный доступ не может считаться отдельным типом атаки. Большинство сетевых атак проводятся ради получения несанкционированного доступа. Чтобы подобрать логин telnet, хакер должен сначала получить подсказку telnet на своей системе. После подключения к порту telnet на экране появляется сообщение «authorizat ion required to use this resource» (для пользования этим ресурсом нужна авторизация). Если после этого хакер продолжит попытки доступа, они будут считаться «несанкционированными». Источник таких атак может находиться как внутри сети, так и снаружи.

Способы борьбы с несанкционированным доступом достаточно просты. Главным здесь является сокращение или полная ликвидация возможностей хакера по получению доступа к системе с помощью несанкционированного протокола. В качестве примера можно рассмотреть недопущение хакерского доступа к порту telnet на сервере, который предоставляет Web-услуги внешним пользователям. Не имея доступа к этому порту, хакер не сможет его атаковать. Что же касается межсетевого экрана, то его основной задачей является предотвращение самых простых попыток несанкционированного доступа.

Вирусы и приложения типа «троянский конь»

Рабочие станции конечных пользователей очень уязвимы длявирусов и троянских коней. Вирусами называются вредоносные программы, которые внедряются в другие программы для выполнения определенной нежелательной функции на рабочей станции конечного пользователя «Троянский конь» - это не программная вставка, а настоящая программа, которая выглядит как полезное приложение, а на деле выполняет вредную роль. Борьба с вирусами и «троянскими конями» ведется с помощью эффективного антивирусного программного обеспечения, работающего на пользовательском уровне и, возможно, на уровне сети. Антивирусные средства обнаруживают большинство вирусов и «троянских коней» и пресекают их распространение. Получение самой свежей информации о вирусах помогает эффективнее бороться с ними. По мере появления новых вирусов и «троянских коней» предприятие должно устанавливать новые версии антивирусных средств и приложений.

## Проблемы безопасности локальных вычислительных сетей и интегрированных информационных систем управления предприятием

Для обеспечения безопасности локальных вычислительных сетей (ЛВС) и ИИСУП необходимы:

■ разработка адекватной политики управления и безопасности ЛВС и ИИСУП в целом;

■ обучение персонала предприятия особенностям использования ЛВС и ее защиты;

■ использование адекватных механизмов защиты для рабочих станций;

■ применение адекватной защиты в ходе передачи информации.

Политика безопасности должна определять роль, каждого служащего в деле ЛВС реализации адекватной защищенности и передаваемой в ней информации. Управление ЛВС и ИИСУП должно иметь необходимые для нормальной работы финансовые средства, время и ресурсы. Слабое управление сетью может приводить к ошибкам защиты. Ответственность за внедрении механизмов и мер защиты, а также за следование правилам использования ПК в среде ЛВС и ИИСУП обычно ложится на пользователей ПК. Поэтому им должны быть даны соответствующие инструкции и рекомендации для поддержания приемлемого уровень защиты в среде ЛВС и ИИСУП.

Неавторизованный доступ к локальной вычислительной сети

Три общих метода используются, чтобы получить не авторизованный доступ: общие пароли, угадывание пароля и перехват пароля. Общие пароли позволяют неавторизованному пользователю получить доступ к ЛВС и привилегии законного пользователя, это делается с одобрения какого-либо законного пользователя, под чьим именем осуществляется доступ. Угадывание пароля является традиционным способом неавторизованного доступа. Перехват пароля - это процесс, в ходе которого за конный пользователь, сам не зная того, раскрывает учетный идентификатор пользователя и пароль. Это может быть выполнено с применения программы троянского коня, которая имеет для пользователя вид нормальной программы входа в ЛВС. Другим методом является перехват пароля и идентификатора пользователя, передаваемых по ЛВС в незашифрованном виде. Методы перехвата открытого трафика ЛВС, включая пароли, широко распространены сегодня. Неавторизованный доступ к ЛВС становится особенно возможным при таких условиях, как:

■ отсутствие или недостаточность схемы идентификации и аутентификации;

■ наличие совместно используемых паролей;

■ плохое управление паролями или применение легких для угадывания паролей;

■ использование известных системных брешей и уязвимых мест, которые не были устранены;

■ эксплуатация однопользовательских ПК, не имеющих парольной защиты во время загрузки;

■ неполное использование механизмов блокировки ПК;

■ наличие хранимых в пакетных файлах на дисках ПК па роли доступа к ЛВС;

■ слабый физический контроль за сетевыми устройствами;

■ незащищенные модемы;

■ отсутствие тайм-аута при установлении сеанса и регистрации неверных попыток;

■ не отключенный терминал при многочисленных неудачных попытках установления сеанса и регистрации таких попыток;

■ отсутствие сообщений «дата/время последнего удачного сеанса» и «неуспешная попытка установления сеанса» в на чале сеанса;

■ отсутствие верификации пользователя в реальном времени (для выявления маскарада).

Несанкционированный доступ к ресурсам локальной вычислительной сети

Несанкционированный доступ происходит, когда пользователь, законный или несанкционированный, получает доступ к ресурсу, который пользователю не разрешено использовать. Несанкционированный доступ случается и потому, что механизм управления доступом или механизм назначения привилегий обладает недостаточной степенью детализации. В этих случаях единственный способ предоставить пользователю необходимые права доступа или привилегии для выполнения определенной функции состоит в том, чтобы предоставлять ему больше привилегий («доступа»), чем необходимо. Несанкционированный доступ к ресурсам ЛВС может происходить из -за:

■ использования при назначении прав пользователям по умолчанию таких системных установок, которые являются «слишком разрешающими» для пользователей;

■ неправильного использования привилегий администратора или менеджера ЛВС;

■ данных, хранящихся с неадекватным уровнем защиты или вообще без защиты;

■ недостаточного или неправильного использования механизма назначения привилегий для пользователей.

Раскрытие данных

Раскрытие данных или программного обеспечения ЛВС происходит, когда информация не зашифрована, или же путем просмотра лицом, которое не имеет доступа к данным экрана монитора или распечаток информации. Компрометация данных ЛВС может происходить, если используются:

■ неправильные установки управления доступом;

■ данные, которые считаются достаточно критичными, чтобы использовать шифрование, но которые тем не менее хранятся в незашифрованной форме;

■ исходные тексты приложений, в незашифрованной форме;

■ мониторы, или станции печати в помещениях, где много посторонних людей;

■ резервные копии данных и программного обеспечения, хранимые в открытых помещениях.

Неавторизованная модификация данных и программ

Неавторизованная модификация данных или программного обеспечения происходит, когда в файле или программе производятся неавторизованные изменения (добавление, удаление или модификации). Когда незаметная модификация данных происходит в течение длительного периода времени, то эти измененные данные могут распространиться по ЛВС и привести к искажению всей базы данных, электронных таблиц и т. д.

Неавторизованная модификация данных и программного обеспечения может происходить, когда:

■ выдается разрешение на запись пользователям, которым требуется лишь доступ по чтению;

■ имеются необнаруженные изменения в программном обеспечении, включая добавление кода для создания программы троянского коня;

■ не обозначена криптографическая контрольная сумма критических данных;

■ имеется механизм привилегий, позволяющий избыточное разрешение записи;

■ отсутствуют средства выявления и защиты от вирусов.

Раскрытие трафика локальной вычислительной сети

Раскрытие трафика ЛВС происходит, когда кто-то, кому это не разрешено, читает информацию, или получает к ней доступ незаконным способом, во время ее передачи через ЛВС. Трафик ЛВС может быть скомпрометирован при прослушивании и перехвате информации, передаваемой по транспортной среде ЛВС (путем подключения к кабелю сети, прослушивания трафика, передаваемого по эфиру, подключения к сети с помощью сетевого анализатора и т. д.). Компрометация трафика ЛВС может происходить если:

■ применена неадекватная физическая защита устройств ЛВС и среды передачи;

■ ведется передача открытых данных с использованием широковещательных протоколов и по среде ЛВС.

Подмена трафика локальной вычислительной сети

Подмена трафика ЛВС означает использование его неавторизованным способом или просто злоумышленником. В этой ситуации возникают возможности: получать сообщение, маскируясь под легитимное место назначения или маскироваться под машину -отправитель и посылать сообщения кому-либо. Подмена трафика ЛВС или его модификация может происходить при:

■ передаче трафика ЛВС в открытом виде;

■ отсутствии отметки даты/времени (показывает время посылки и время получения);

■ отсутствии механизма кода аутентификации сообщения или цифровой подписи;

■ отсутствии механизма аутентификации в реальном масштабе времени (для защиты от воспроизведения).

Разрушение функций локальной вычислительной сети

Разрушение функциональных возможностей ЛВС наступает в результате:

■ неспособности своевременно выявить ненормальный режим функционирования трафика, допустим, его чрезмерную переполненность;

■ неспособности в нужный момент перенаправить трафик, выявить отказы аппаратных средств ЭВМ и т. д.;

■ использования такой конфигурации ЛВС, когда становятся вполне возможными отказы элементов этой сети в том или ином месте;

■ неавторизованных изменений компонентов аппаратных средств ЭВМ (переконфигурирование адресов на автоматизированных рабочих местах, изменение конфигурации маршрутизаторов или хабов и т. д.);

■ неправильного обслуживания аппаратных средств ЛВС;

■ недостаточной физической защиты аппаратных средств ЛВС.

## Оценка рисков

От того, насколько полно и правильно будет проанализирована защищенность информационных и других ресурсов предприятия, зависит эффективность всей его деятельности.

Точную оценку защищенности ресурсов получить невозможно, но можно уяснить наиболее уязвимые места и участки и сделать прогноз о возможных угрозах. При анализе рисков принимаются во внимание и учитыаются:

■ модель нарушителя;

■ защита информации в компьютерных системах;

■ защита объекта от несанкционированного доступа;

■ защита речевой информации;

■ экспертная оценка защищенности информационных и других ресурсов на предприятии;

■ экспертная оценка организационно-штатной структуры предприятия;

■ экспертная оценка нормативно-распорядительной документации в области безопасности.

# ТРЕБОВАНИЯ ПО ОБЕСПЕЧЕНИЮ КОМПЛЕКСНОЙ СИСТЕМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

***Требования по обеспечению информационной безопасности корпоративной информационной системы предприятия***

• Организационные меры защиты, сводящиеся к регламентации;

- допуска к использованию информационных ресурсов;

- процессов проведения технологических операций с информационными ресурсами КИС;

- процессов эксплуатации, обслуживания и модификации аппаратных и программных ресурсов.

• Технические меры и методы защиты, заключающиеся в применении средств;

- идентификации и аутентификации;

- разграничения доступа;

- обеспечения и контроля целостности;

- антивирусной защиты;

- межсетевого экранирования;

- контроля электронной почты;

- контроля и регистрации событий безопасности;

- криптографической защиты;

- виртуальных частных сетей (VPN);

- централизованного управления системой информационной безопасности.

***Требования к программно-аппаратным средствам.***

Программно-аппаратные средства ИС предприятия должны:

• обеспечивать возможность комплексного решения вопросов защиты информации в том числе возможность удаленного администрирования;

• иметь встроенные средства криптографической защиты информации и поддерживать технологию электронной подписи;

• полноценно функционировать в распределенной корпоративной среде и иметь минимальное программное обеспечение на клиентских местах;

• учитывать требования действующих нормативных документов по защите информации АС, а также законодательство, ГОСТы, нормативные, руководящие и другие документы уполномоченных государственных органов по вопросам информатизации, защиты информации и обеспечения информационной безопасности.

***Требования к подсистеме идентификации и аутентификации.***

Эта подсистема должна гарантировать:

• невозможность администрирования для непривилегированного пользователя;

• возможность разграничения доступа по группам пользователей, их местоположению, времени обращения;

• осуществление аутентификации перед сменой пароля;

• отслеживание неудачных попыток входа в систему, задержек после ввода неверного пароля перед следующей попыткой, оперативное оповещение администратора безопасности при нескольких последовательных неудачных попытках входа в систему;

• системную защиту данных, использующихся для аутентификации, и регистрационных данных пользователей;

• проверку требований к паролям (по длине, допустимым символам и т. п.); ограничение на доступ к системной базе паролей и на их отображение на экране;

• защиту данных, используемых при аутентификации, и хранение паролей только в зашифрованном виде;

• обязательность периодической смены паролей, а также отличий новых паролей от старых;

• обязательную аутентификацию пользователей при до ступе к базе данных, смену стандартных паролей, используемых при установке;

• демонстрацию при входе пользователя в систему - времени последнего входа/выхода, использовавшихся сервисах, числе неудачных попыток входа с данным именем после последнего сеанса.

***Требования к подсистеме управления доступом.***

Для управления доступом используются:

■ безопасный сервер аутентификации для протоколирования верификации попыток аутентификации и их результатов;

■ идентификатор пользователя и права доступа на чтение, запись, выполнение программ;

■ профиль пользователя;

■ его подразделение.

При этом также необходимо, чтобы:

■ были определены правила доступа, основанные на атрибутах доступа, и правила доступа по умолчанию;

■ доступ к устройствам ввода/вывода был регламентирован

административными и программно-техническими мерами;

■ в базе данных были определены атрибуты доступа для объектов и субъектов (для объектов атрибуты должны устанавливаться в процессе операций импорта/экспорта);

■ правила доступа распространялись только на пользователей, прошедших авторизацию;

■ существовали утвержденные списки управления доступом или правила управления доступом;

■ различающиеся права доступа для объектов и субъектов были проверены на согласованность;

■ в базе данных была обеспечена защита от чтения и модификации информации, относящейся к политике безопасности, в частности: данных, относящихся к идентификации и аутентификации. Причем, пользователи должны иметь возможность в любой момент закрыть (приостановить) свой сеанс и возобновить его после повторной аутентификации; точек входа и соответствующих им параметров (системных и пользовательских); атрибутов, относящихся к политике безопасности и устанавливаемых по умолчанию.

***Требования к подсистеме протоколирования аудита.***

Данные, относящиеся к протоколу (регистрационный журнал), должны быть защищены от изменения. С этой целью:

• система должна позволять идентифицировать и показывать текущие события;

• события, которые могут привести к нарушению целостности регистрационного журнала, должны быть перечислены в документации администратора безопасности;

• должно быть обеспечено протоколирование доступа к базе данных;

• события, подлежащие регистрации, устанавливаются для пользователей, групп пользователей, объектов базы данных;

• действия пользователей с полномочиями администраторов подвергаются аудиту на предмет адекватности текущей ситуации;

• средства протоколирования/аудита должны иметь возможность отслеживать события следующих классов:

- использование механизмов идентификации и аутентификации;

- помещение объектов в адресное пространство пользователя;

- создание, модификация, удаление объектов; а также действия привилегированных пользователей;

- передача данных за пределы системы;

- начало и окончание работы системы, сеансов, точки входа;

• модификация прав доступа и привилегий.

***Требования к подсистеме защиты повторного использования объектов.***

Они нижеследующие:

- вся информация, относящаяся к атрибутам безопасности и авторизации, не содержащаяся в объекте, должна быть выгружена из памяти (уничтожена) после выгрузки объекта;

- данные (включая зашифрованные) относящиеся к незагруженному объекту, должны быть недоступны для любых других объектов, включая и те, что используют незагруженный объект.

***Требования к защите критичной информации.***

Их также немного:

- когда пользовательский сеанс приостановлен (блокирован), вывод также должен быть приостановлен, экран монитора погашен;

- база данных должна иметь возможность перед процедурой инициализации сеанса выдать пользователю предупреждение об ограничениях, связанных с текущей ситуацией.

***Требования к средствам обеспечения целостности.***

Их значительно больше, и они разнообразнее:

■ процедуры (решаемые задачи) должны быть документированы; в частности, должны быть рассмотрены вопросы восстановления в случае сбоев оборудования и нарушения целостности данных;

■ после успешного входа в систему пользователь должен получить следующую информацию:

■ кто последний раз входил в систему (пользователь, процесс и т. п.);

■ дату и время последнего, успешного входа/выхода в систему/из системы;

■ сервис, который был использован во время сеанса;

■ число неудачных попыток входа в систему после завершения последнего сеанса;

■ данные о пользовательском идентификаторе;

База данных должна быть в состоянии ограничить возможности незарегистрированного пользователя в попытках входа в систему посредством:

* использования задержки после неудачной попытки входа в систему;
* блокирования доступа к данным пользователя, не вошедшего в систему.

База данных должна быть в состоянии работать в нормальном режиме и режиме технического обслуживания (технологическом):

■ по умолчанию для пользователей должны быть недоступны каталоги, созданные другими пользователями и программами;

■ функции администратора, связанные с обеспечением информационной безопасности, должны быть недоступны прочим пользователям и процессам;

■ процедура восстановления данных должна обеспечивать восстановление данных всех категорий вплоть до минимального уровня защищенности;

■ процедуру восстановления можно проводить только в режиме технического обслуживания.

Обеспечение безопасности модификации и получения информации достигается благодаря тому, что:

- обычные пользователи не могут перевести систему из нормального режима в режим технического обслуживания;

- в режиме технического обслуживания обычные пользователи не имеют доступа в систему;

- база данных ведет отчетность по каждому пользователю отдельно.

***Требования к средствам управления информационной безопасности.***

Надежность средств управления безопасностью обеспечивается не в последнюю очередь разделением ролей и обязанностей администраторов. При этом должны присутствовать как минимум администратор безопасности, системный администратор, пользователи.

Особо контролироваться должны вопросы перераспределения и добавления должностных обязанностей, связанных с ИБ; что касается средств администрирования, относящихся к ИБ, то они должны контролироваться на предмет их несанкционированного использования, модификации, уничтожения.

В системе должны также использоваться механизмы защиты при регистрации новых пользователей, а в базе данных должны присутствовать механизмы защиты, обеспечивающие:

- невозможность включать/выключать механизмы защиты;

- выбирать или изменять события, подлежащие протоколированию/аудиту;

- изменять установленные по умолчанию события и атрибуты защиты. Требования к средствам управления ИБ диктуют в том числе и:

- установление предельного времени пассивности пользователей, после которого они исключаются из числа легальных пользователей;

- возможность для системного администратора проводить аудит действий одного или выбранной группы пользователей;

- защита средств проведения аудита от неавторизованного использования, модификации, уничтожения;

- обязательное наличие в базе данных при установке механизма выбора и обновления параметров конфигурации;

- обеспечение защиты механизма регистрации параметров пользователя от несанкционированного удаления, модификации, ознакомления;

- существование механизма удаления пассивных пользователей;

- обеспечение возможности для администратора отменить команду удаления пользователя из списков;

- существование защитного механизма, обеспечивающего доступ только авторизованного персонала к выполнению функций администратора.

***Требования к Межсетевому Экрану.***

МЭ должен обеспечивать:

- фильтрацию на сетевом уровне. Решение по фильтрации может приниматься для каждого сетевого пакета независимо на основе, по крайней мере, сетевых адресов отправите ля и получателя или на основе других эквивалентных атрибутов;

- фильтрацию пакетов служебных протоколов, служащих для диагностики и управления работой сетевых устройств;

- фильтрацию с учетом входного и выходного сетевого интерфейса как средство проверки подлинности сетевых адресов;

- фильтрацию с учетом любых значимых полей сетевых пакетов;

- фильтрацию на транспортном уровне запросов на установление виртуальных соединений. При этом, по крайней мере, учитываются транспортные адреса отправителя и получателя;

- фильтрацию на прикладном уровне запросов к прикладным сервисам. В этом случае, по крайней мере, учитываются прикладные адреса отправителя и получателя;

- фильтрацию с учетом даты/времени.

К дополнительным требованиям МЭ можно отнести:

* обеспечение процесса идентификации и аутентификации
* обеспечение регистрации и учет запросов на установление виртуальных соединений,
* обеспечивать контроль целостности программной и информационной части МЭ
* обеспечивать локальную сигнализацию попыток нарушения правил фильтрации
* должен содержать средства контроля за целостностью своей программной и информационной части
* иметь процедуры восстановления
* обеспечивать возможность регламентного тестирования

# СИСТЕМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

## Системы обнаружения атак.

Intrusion Detective System (IDS) - система обнаружения атак в реальном времени. Принцип ее действия довольно прост: в систему вложена база известных багов различного типа. При коннекте хакера IDS просматривает входящий траффик на предмет вредоносных действий (кода, команд и т.д.). Если она находит такие действия, то сразу же сообщает об этом администратору или лицу, ответственному за безопасность (выводит сообщение на консоль, шлет почту, пишет в лог, шлет на пейджер/SMS и т.д.). Использование стандартных/классических багов (к примеру, SQL-инъекция) дает возможность атакующему выполнить произвольный запрос к базе данных – прочитать содержимое любых таблиц, удалить, изменить или добавить данные; получить возможность чтения и/или записи локальных файлов и выполнения произвольных команд на атакуемом сервере.

Основными свободно распространяемыми СОА являются:

1) Snort;

2) Bro;

3) Prelude;

4) OSSEC;

5) STAT.

Snort выполняет протоколирование, анализ, поиск по содержимому, а также широко используется для активного блокирования или пассивного обнаружения целого ряда нападений и зондирований, таких как переполнение буфера, стелс-сканирование портов, атаки на веб-приложения, SMB-зондирование и попытки определения ОС. Специальные модули производят дефрагментацию IP, декодирование HTTP-запросов.

Система Bro является открытой и распространяется по собственной открытой лицензии в исходных текстах и бинарных пакетах для ряда UNIX-платформ. Обнаружение атак выполняется на нескольких уровнях: входящий сетевой трафик разбирается для выявления семантики уровня приложений, после чего полученная трасса событий прикладного уровня анализируется набором событийно-ориентированных анализаторов и сравнивается с шаблонами атак. Система Bro является сетевой системой обнаружения атак. Она представляет собой набор модулей декомпозиции данных различных сетевых протоколов (от сетевого до прикладного уровня) и набор сигнатур над событиями соответствующих протоколов. Сигнатуры Bro фактически представляют собой регулярные выражения в алфавитах протоколов.

Prelude представляет собой полнофункциональную, т.е. гибридную(контролируются не только события, происходящие в сети, но и то, что творится на локальных компьютерах, что повышает вероятность обнаружения попытки вторжения), СОА. Она разрабатывалась, прежде всего, для работы на компьютерах под управлением GNU/Linux. Может быть сконфигурирован как для просмотра локальных log-файлов, так и работать в сетевом режиме, контролируя поступающие по сети данные. В качестве источника информации, использует различные журналы регистрации.

OSSEC – представитель систем обнаружения вторжений на уровне хоста, позволяющий анализировать журналы, проверять целостность файлов и реестра, обнаруживать rootkit и активно реагировать на попытки взлома системы. В функциональность заложена возможность анализировать локальные журналы как уровня ОС, так и отдельных приложений. OSSEC обрабатывает информацию о событиях, которая приходит от других систем, от узлов с установленными агентами OSSEC, от межсетевых экранов, сетевых СОА. На основании правил в формате XML выдаются оповещения о подозрительных действиях. OSSEC может отслеживать разрешения, размер и владельца файла. Сохраняет значения хэш-функций MD5 и SHA1 для файлов, за которыми было указано наблюдать. База данных всех этих значений хранится на сервере.

Система STAT включает в себя набор компонентов обнаружения атак различных уровней – сетевой (NetSTAT), узловой (USTAT, WinSTAT), приложений (WebSTAT), т.е. является классической гибридной системой. Исследования по проекту STAT сосредоточены на обнаружении вторжений в реальном масштабе времени. Для этого анализируются состояние систем и процессы перехода в них. Основная идея заключается в том, что некоторые последовательности действий, однозначно указывающие на присутствие нарушителя, переводят систему из начального (санкционированного) состояния в несанкционированное.

Большинство централизованных систем обнаружения вторжений определяют факт вторжения на основе «контрольного следа». STAT фильтрует и обобщает эту информацию. Результаты, преобразованные в удобный для анализа вид, называются сигнатурами и представляют собой важнейший элемент в подходе STAT. Последовательность действий, описанная сигнатурой, переводит систему через ряд состояний к несанкционированному виду. Вторжение определяется по переходам между состояниями, которые зафиксированы в наборах продукционных правил.

Одно из преимуществ данного подхода состоит в том, что нападение может быть выявлено еще до того, как система окажется скомпрометированной, и соответственно противодействие начнется раньше.

Еще одним примером систем обнаружения атак в реальном времени служит система обнаружения атак RealSecure. Она построена по технологии анализа сетевых пакетов в реальном масштабе времени (real-time packet analysis) относится к системам обнаружения атак, ориентированных на защиту целого сегмента сети (network-based). Для защиты конкретного узла (Host-based) корпоративной сети может применяться система RealSecure.

Как только атака распознается, происходит оповещение администратора через консоль управления или электронную почту. Кроме того, атака может быть зарегистрирована в базе данных, а также все операции при осуществлении атаки могут быть записаны для дальнейшего воспроизведения и анализа. В случае осуществления атаки, которая может привести к выведению из строя узлов. RealSecure позволяет обнаруживать большое число атак и иных контролируемых событий:

- "Отказ в обслуживании"

- "Неавторизованный доступ"

- "Предварительные действия перед атакой"

- "Подозрительная активность" (Suspicious activity)

- "Анализ протокола" (Protocol decode)

- "Детектор атак" (Attack Detector)

- "Анализатор протоколов" (Protocol Analyzer)

- "Web-сторож" (Web Watcher)

- "Windows-сети" (For Windows Networks).

- "Запись сессий" (Session Recorder).

- "Модуль слежения в DMZ" (DMZ Engine).

- "Модуль слежения до межсетевого экрана"

Возможность адаптивности у данных СОА практически отсутствует (за исключением существования модуля статистического анализа у СОА Snort).

Следовательно, актуальной является задача разработки гибридной адаптивной СОА. Ее обобщенная модель может быть представлена в следующем виде (см. рисунок):



Рис.1. Обобщенная модель гибридной адаптивной СОА

Она состоит из модуля управления, осуществляющего управление системой. Мониторинг осуществляется как на системном, так и на сетевом уровнях. За это отвечают модули системного и сетевого мониторинга. Полученные от этих модулей данные анализируется с помощью нейросетей, что позволит системе адаптироваться к новым атакам и новым типам известных атак.

### Системы обнаружения атак в режиме отложенного анализа

Апостериорный (отложенный) анализ может использоваться со следующей целью:

− расследование информационных преступлений и инцидентов;

− выявление атак, не являющихся информационным преступлением (сбор информации об инфраструктуре сети, сканирование портов и пр.);

− сбор информации об уязвимостях информационной системы с целью их устранения;

− анализ активности отдельных пользователей;

 − минимизация системных требований к СОА.

В отложенном анализе разыскивается уязвимое ПО, ошибки в конфигурациях и т.д. Системы отложенного анализа проверяют версии работающих в системе приложений на наличие известных уязвимостей и слабых паролей, проверяют содержимое специальных файлов в директориях пользователей или проверяют конфигурацию открытых сетевых сервисов.

 В отложенном режиме реализация атаки на защищаемые информационные и вычислительные ресурсы не будет замечена вовремя и уж тем более не будет отражена с помощью имеющихся средств защиты. В данном режиме СОА может быть использована в лучшем случае как средство журналирования всех этапов атаки и последующей криминалистической экспертизы.

Системы отложенной обработки анализируют содержимое журналов регистрации событий или массив предварительно записанного трафика, а системы реального времени — входящий поток событий от программных датчиков. Очевидно, что адекватное реагирование на попытку реализации атаки, включая ее предотвращение, возможно только при использовании систем реального времени. В то же время это не означает, что СОА реального времени «лучше», чем системы отложенной обработки. Так, СОА реального времени, не имеющая функций по предотвращению атак, заведомо менее эффективна, чем аналогичная система с отложенной обработкой, поскольку в системе реального времени одним из основных критериев эффективности является простота используемых алгоритмов, а не их оптимальность с позиций надежности обнаружения атак. Поэтому выбор того или иного типа СОА должен делаться исходя из анализа задач, которые ставятся перед системой обнаружения.

### Методы обхода соа

Системы обнаружения атак, помимо достоинств, имеют и недостатки.

1) Безопасность против производительности сети

Администраторы безопасности в случае IDS оказываются перед дилеммой. Если корпоративную сеть надо защитить изнутри целиком, то в наиболее важных точках сети потребуется установить датчики и предоставить им достаточное количество вычислительных ресурсов, чтобы можно было принимать и фильтровать все поступающие пакеты данных. При тотальном мониторинге посредством датчиков увеличивается нагрузка на сеть. Помимо прочего, взаимодействие между компонентами IDS порождает дополнительный трафик в корпоративной сети, и тогда при высоких требованиях к безопасности не обойтись без частичной перепланировки сети.

2) Безопасность против рентабельности

Одни только издержки на приобретение всех необходимых компонентов IDS могут вызвать сомнение в рентабельности такого подхода к безопасности.

С учетом стоимости перепланировки сети и последующих затрат на администрирование и обслуживание финансовый болевой порог будет достигнут незамедлительно. Причем даже при максимальных затратах системы IDS, как правило, не предлагают никакой защиты от неизвестных типов атак.

Рассмотрим некоторые методы обхода IDS.

1. Таймаут IDS меньше времени сборки у Жертвы.

Сценарий: Предположим, что на Системе обнаружения таймаут составляет 15 секунд, а на компьютере, на который нацелился хакер - 30 секунд, это дефолтное время для ОС Linux. Как показано на рисунке, после отсылки первого фрагмента атакующий в промежутке от 15 до 30 секунд может послать второй.



Рис. 2. Таймаут IDS

Очевидно, что у жертвы поток собирается правильно, в то время как IDS отбрасывает первый пакет из-за превышения времени таймаута (и второй соответственно не принимается). Результат - тревога не подымается.

2. Таймаут IDS больше чем у Жертвы.

Сценарий: Как уже говорилось, по умолчанию таймаут у Snort-а 60 секунд, а у Linux/FreeBSD - 30. Это можно использовать. Как показано ниже, атакующий делит пакет на 4 сегмента: 1, 2, 3, 4.



Рис. 3. Таймаут IDS

В данном случает фрагменты 2 и 4 не несут никакой информации (2' и 4'), они получаются как жертвой, так и IDS. Атакующий ожидает, пока пакеты на компьютере Жертвы будут сброшены. Прелесть этой атаки в том, что у жертвы пока так и нету фрагмента 1 и, по правилам, ICMP сообщение не будет отослано. Атакующий посылает новые пакеты (1 и 3) с правильной, нужной ему информацией. На этом этапе жертва имеет только фрагменты

1 и 3. Если теперь нападающий отошлет пакеты 2 и 4 не с мусором, а с нужной ему информацией, то они благополучно дойдут до жертвы, в то время как IDS получит только их, так как предыдущие части были собраны и выкинуты за ненадобностью. Результат - Жертва имеет все 4 части с правильной информацией, IDS только половину.

Атака IDS с использованием TTL

В этом случае нападающему необходимо знать топологию сети, сколько роутеров находится между ним и Жертвой. Выяснить это можно, например, утилитой traceroute. Обратимся к рисунку:



Рис.4. Использование traceroute.

Ясно, что между IDS и Жертвой находится роутер. Атакующий разбивает свою атаку на три части, три фрагмента. Первому устанавливается большое время TTL и он получается как компьютером жертвы, и IDS. Второй (2') является фальшивым и имеет TTL = 1, он приходит на IDS, но дальше роутера не уходит, так как его время жизни уже закончилось. Затем в дело идет фрагмент 3 с правильным временем жизни, IDS собирает все три фрагмента правильно, в то время как Жертва ждет середину цепочки. Наконец в дело идет правильный пакет 2 и атака удачно завершается. Результат - IDS обманута, у нее только 3 фрагмент, а жертва (вероятно) выведена из строя.

**Использование IDS помогает достичь нескольких целей:**

* Обнаружить вторжение или сетевую атаку;
* Спрогнозировать возможные будущие атаки и выявить уязвимости для предотвращения их дальнейшего развития. Атакующий обычно выполняет ряд предварительных действий, таких как, например, сетевое зондирование (сканирование) или другое тестирование для обнаружения уязвимостей целевой системы;
* Выполнить документирование существующих угроз;
* Обеспечить контроль качества администрирования с точки зрения безопасности, особенно в больших и сложных сетях;
* Получить полезную информацию о проникновениях, которые имели место, для восстановления и корректирования вызвавших проникновение факторов;
* Определить расположение источника атаки по отношению к локальной сети (внешние или внутренние атаки), что важно при принятии решений о расположении ресурсов в сети.

### Архитектура IDS

**Архитектура IDS:**

* Сенсорную подсистему, предназначенную для сбора событий, связанных с безопасностью защищаемой сети или системы;
* Подсистему анализа, предназначенную для выявления сетевых атак и подозрительных действий;
* Хранилище, в котором накапливаются первичные события и результаты анализа;
* Консоль управления, позволяющая конфигурировать IDS, наблюдать за состоянием защищаемой системы и IDS, просматривать выявленные подсистемой анализа инциденты.

По способам мониторинга IDS системы подразделяются на network-based (NIDS) и host-based (HIDS).

Основными коммерческими IDS являются network-based. Эти IDS определяют атаки, захватывая и анализируя сетевые пакеты. Слушая сетевой сегмент, NIDS может просматривать сетевой трафик от нескольких хостов, которые присоединены к сетевому сегменту, и таким образом защищать эти хосты.

**Преимущества NIDS:**

* Большое покрытие для мониторинга и в связи с этим централизованное управление;
Несколько оптимально расположенных NIDS могут просматривать большую сеть.
* Не влияют на производительность и топологию сети.
NIDS обычно являются пассивными устройствами, которые прослушивают сегменты сети без воздействия на её нормальное функционирование. Таким образом, обычно бывает легко модифицировать топологию сети для размещения таких IDS.

**Недостатки NIDS:**

* Обладают высокой ресурсоёмкостью;
Для NIDS может быть трудно обрабатывать все пакеты в большой или занятой сети, и, следовательно, они могут пропустить распознавание атаки, которая началась при большом трафике.
* Требуют дополнительной настройки и функциональности сетевых устройств;
Например, многие коммутаторы, на которых построены сети, не предоставляют универсального мониторинга портов, и это ограничивает диапазон мониторинга сенсора NIDS только одним хостом. Даже когда коммутаторы предоставляют такой мониторинг портов, часто единственный порт не может охватить весь трафик, передаваемый коммутатором.
* Не могут анализировать зашифрованную информацию;
Эта проблема возрастает, чем больше организации (и атакующие) используют VPN.
* Не могут распознать результат атаки;
NIDS не могут сказать была ли атака успешной, они могут только определить, что атака была начата. Это означает, что после того как NIDS определит атаку, администратор должен вручную исследовать каждый атакованный хост для определения, происходило ли реальное проникновение.
* Некоторые NIDS имеют проблемы с определением сетевых атак, которые включают фрагментированные пакеты.
Такие фрагментированные пакеты могут привести к тому, что IDS будет функционировать нестабильно.

Host-based IDS имеют дело с информацией, собранной внутри единственного компьютера. Такое выгодное расположение позволяет HIDS анализировать деятельность с большой достоверностью и точностью, определяя только те процессы и пользователей, которые имеют отношение к конкретной атаке в ОС. НIDS обычно используют информационные источники двух типов: результаты аудита ОС и системные логи.

**Преимущества HIDS:**

* Имеют возможностью следить за событиями локально относительно хоста, могут определять атаки, которые не могут видеть NIDS;
* Могут функционировать в окружении, в котором сетевой трафик зашифрован;
Это становится возможным, когда host-based источники информации создаются до того, как данные шифруются, и/или после того, как данные расшифровываются на хосте назначения.
* Не требуют дополнительной функциональности сетевых устройств.
Например, на функционирование HIDS не влияет наличие в сети коммутаторов.

**Недостатки HIDS:**

* Не имеют централизованного управления;
HIDS более трудны в управлении, так как они должны быть сконфигурированы и управляться для каждого целевого хоста.
* Могут быть блокированы некоторыми DoS-атаками или даже запрещены;
Так как по крайней мере источники информации (сенсоры) или часть средств анализа для HIDS расположены на том же хосте, который является целью атаки, то, как составная часть атаки, IDS может быть атакована и запрещена.
* Обладают высокой ресурсоёмкостью;
HIDS используют вычислительные ресурсы хостов, за которыми они наблюдают, что влияет на производительность наблюдаемой системы.
* Малое покрытие для мониторинга.
HIDS не полностью соответствуют возможности определения сканирования сети или других аналогичных исследований, когда целью является вся сеть, так как IDS наблюдает только за сетевыми пакетами, получаемыми конкретным хостом.

По способам определения вредоносного трафика IDS системы подразделяются на: signature-based (сигнатурного метода), anomaly-based (метода аномалий) и policy-based (метода, основанного на политике).

Детекторы атак анализируют деятельность системы, используя для этого событие или множество событий на соответствие заранее определенному образцу, который описывает известную атаку. Соответствие образца известной атаке называется сигнатурой, определение атаки или вторжения иногда называют "сигнатурным определением".

**Преимущества сигнатурного метода:**

* Эффективное определение атак и отсутствие большого числа ложных сообщений;
* Надежная диагностика использования конкретного инструментального средства или технологии атаки.

Это позволяет администраторам, независимо от уровня их квалификации в области безопасности, начать процедуры обработки инцидента, а также скорректировать меры обеспечения безопасности.

**Недостатки сигнатурного метода:**

* Обязательное обновление базы данных для получения сигнатур новых атак.

Метод аномалий состоит в определении ненормального (необычного) поведения на хосте или в сети. Детекторы аномалий предполагают, что атаки отличаются от "нормальной" (законной) деятельности и могут, следовательно, быть определены системой, которая умеет отслеживать эти отличия. Детекторы аномалий создают профили, представляющие собой нормальное поведение пользователей, хостов или сетевых соединений. Эти профили создаются, исходя из данных истории, собранных в период нормального функционирования. Затем детекторы собирают данные о событиях и используют различные метрики для определения того, что анализируемая деятельность отклоняется от нормальной.

**Преимущества метода аномалий:**

* Определение атаки без знания конкретных деталей (сигнатуры);
* Детекторы аномалий могут создавать информацию, которая в дальнейшем будет использоваться для определения сигнатур атак.

**Недостатки метода аномалий:**

* Большое количество ложных сигналов при непредсказуемом поведении пользователей и непредсказуемой сетевой активности;
* Временные затраты на этапе обучения системы, во время которого определяются характеристики нормального поведения.

Метод, основанный на политике (policy-based) заключается в написании правил сетевой безопасности.в терминах распределения доступа, например какие сети могут взаимодействовать друг с другом и какие протоколы при этом могут использоваться.

**Преимущества метода политик:**

* Имеет преимущество при обнаружении новых (неизвестных) атак.

**Недостатки метода политик:**

* Трудоёмкость создания базы политик.

## Система предотвращения вторжений

**Система предотвращения вторжений** ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Intrusion Prevention System*) — программная или аппаратная система сетевой и компьютерной безопасности, обнаруживающая вторжения или нарушения безопасности и автоматически защищающая от них.

Системы IPS можно рассматривать как расширение [Систем обнаружения вторжений](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BE%D0%B1%D0%BD%D0%B0%D1%80%D1%83%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D1%82%D0%BE%D1%80%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) (IDS), так как задача отслеживания атак остается одинаковой. Однако, они отличаются в том, что IPS должна отслеживать активность в реальном времени и быстро реализовывать действия по предотвращению атак. Возможные меры — блокировка потоков трафика в сети, сброс соединений, выдача сигналов оператору. Также IPS могут выполнять дефрагментацию пакетов, переупорядочивание пакетов TCP для защиты от пакетов с измененными SEQ и ACK номерами.

Как правило, при рассмотрении IPS "как явления" применяют классификацию, унаследованную от систем обнаружения вторжения – деление продуктов на "сетевые" и "хостовые".

Сетевая IPS представляет некоторое устройство, находящееся на пути передачи сетевого трафика. Основная задача сетевой IPS – защита группы хостов сети от возможных атак путем анализа передаваемого трафика и блокирования трафика, связанного с проведением атак. Хостовая IPS располагается на конкретном хосте и обеспечивает его защиту от разрушающих воздействий путем анализа сетевого трафика, поведения приложений, активируемых системных вызовов и т.п.

При этом эпоха, когда для обнаружения факта атаки считалось достаточным найти некий характерный шаблон трафика, или "сигнатуру", также канули в лету. Считается, что для успешного обнаружения атак современная IPS должна обладать следующими свойствами и функциями:

Во-первых, проводить анализ сеанса взаимодействия с учетом протоколов, используемых для передачи данных (особенно актуально для протоколов прикладного уровня модели OSI).

Во-вторых, выполнять восстановление фрагментированных IP-пакетов до их анализа, не передавать фрагменты IP-датаграмм без проверки.

В-третьих, отслеживать попытки создания перекрывающихся фрагментов IP-датаграмм, попытки перезаписи содержимого TCP-сегментов и предотвращать их.

В-четвертых, обладать знаниями о топологии защищаемой сети и, возможно, поддерживать механизмы нормализации IP/TCP-трафика для предотвращения атак на IPS/IDS классов "insertion" или "evasion".

В-пятых, обеспечивать проверку соответствия логики/форматов работы по протоколу соответствующим RFC.

В-шестых, выполнять статистический анализ данных.

В-седьмых, поддерживать механизмы сигнатурного поиска.

А также обладать возможностью обучения и самообучения.

Кроме того, поскольку IPS может принимать решения о блокировании трафика, необходимо обеспечить надежное и безопасное удаленное управление IPS. Средства конфигурирования IPS должны быть удобны конечным пользователям. Большинство IPS поддерживают возможность задания "пользовательских" правил обнаружения вторжений для возможности подстройки IPS под конкретную среду/требования конкретного заказчика.

Особенностями IPS являются:

- IPS - это in-line устройство на пути трафика, оно прозрачно, то есть не имеет MAC- и IP-адресов портов в сегменте, не использует таблицы коммутации/маршрутизации как коммутаторы/маршрутизаторы;

- проверяются все пакеты, проходящие сквозь него;

- осуществляется проверка всего пакета;

- отслеживаются потоки/сессии;

- принимаются меры в реальном времени (блокировать, пропустить, ограничить скорость);

- блокируются DoS (Denial of Service – отказ в обслуживании) и DDoS (Distributed Denial of Service – распределенный отказ в обслуживании) атаки;

- возможно обнаружение любого типа трафика – разрешенный трафик, шпионские приложения, Peer-to-peer и т.п.;

- могут применяться фильтры, аналогичные фильтрам брандмауэра.

Системs IPS позволяют анализировать сетевой трафик на 2 — 7 уровнях модели OSI, выявлять нежелательную активность и предпринимать соответствующие меры: блокировать источник трафика, отсылать администраторам сообщения с информацией об атаках, журналировать проходящий трафик для ручного анализа и т.д

Системы предотвращения вторжений принято подразделять на сетевые системы — NIPS (англ. Network-based Intrusion Prevention System) и персональные — HIPS (англ. Host-based Intrusion Prevention System). Сетевые системы используются непосредственно в сети организации или предприятия, позволяя анализировать трафик, проходящий в интересующем направлении. Персональные системы, по сути программное обеспечение, устанавливаемое на рабочие станции или сервера, позволяющее контролировать деятельность приложений, а также отслеживать сетевую активность на наличие возможных атак. Как правило, подобные приложения тесно интегрируются с персональными брандмауэрами и антивирусами, устанавливаемыми на те же компьютеры.

Главными вопросами, предваряющими развертывание и внедрение данной системы, наверняка станут эффект данного решения и круг решаемых им проблем. Базовые задачи, решаемые устройствами данного класса, можно разделить на три основных направления:

1. Защита от атак, направленных на уязвимости в приложениях, не закрытые на данный момент производителями программного обеспечения — так называемые атаки Day Zero.

2. Защита от сетевых вирусов-червей, распространяющихся через Интернет.

3. Блокирование нежелательных типов трафика, при перекрытии которых необходим глубокий анализ данных на сетевых уровнях L2-L7.

Рассмотрим каждое направление в отдельности.

После обнаружения опасной уязвимости производители IPS, как правило, в кратчайшие сроки выпускают обновления сигнатур для своих систем с «отпечатками» сетевой активности эксплойтов, использующих данную уязвимость. Злоумышленник, попытавшийся воспользоваться этой брешью, будет остановлен системой предотвращения вторжений, которая заблокирует его трафик заблаговременно, до достижения им атакуемого сервера или рабочей станции. Таким образом, даже в отсутствие необходимых обновлений программного обеспечения, наша инфраструктура будет защищена от внешних атак, направленных на самые последние выявленные уязвимости.

Система предотвращения вторжений, становится для червя непреодолимым барьером. Система ведет анализ проходящего через нее трафика и в случае обнаружении всплеска подозрительного трафика, вроде сканирования различных IP-диапазонов на предмет наличия рабочих станций или открытых портов, блокирует его источник. Таким образом, сетевой червь лишается возможности найти потенциальные жертвы для дальнейшего заражения и, соответственно, перестает распространяться.

Проблема блокирования некоторых нежелательных типов трафика стоит в том или ином виде перед многими компаниями и организациями. Если раньше считалось, что для решения проблемы достаточно закрыть определенный TCP- или UDP-порт, то ныне даже широко распространенное пользовательское программное обеспечение может использовать сложные алгоритмы обмена информацией, что делает задачу блокирования трафика определенных приложений неразрешимой с помощью одного только брандмауэра.

Современные системы предотвращения вторжений анализируют трафик на основе:

• Имеющихся в системе сигнатур.

• Аномалий в потоках данных или используемых протоколах.

Базовым методом работы системы предотвращения вторжения является сигнатурный. Он позволяет с высокой степенью точности отслеживать нелегитимный трафик. Сигнатуры, используемые системой предотвращения вторжений, выпускаются производителем данного устройства и регулярно обновляются. Кроме того, многие системы предотвращения вторжений позволяют создавать и настраивать пользовательские сигнатуры для конкретной настройки системы под различные возможные угрозы. В случае анализа аномалий в трафике, система выявляет подозрительный трафик, характерный для сетевых червей и вирусов и не свойственный для данной сети, и блокирует его. Для этого система некоторое время проходит обучение, анализируя и запоминая типы и характеристики трафика в конкретной сети, после чего переводится в активный режим для выявления аномалий передачи данных.

Современные системы предотвращения вторжений имеют широкий спектр возможных ответных реакций на нелегальный трафик: посылка сигналов тревоги системным администраторам, журналирование подозрительного трафика, блокирование источника трафика, сброс TCP-сессии, ограничение полосы канала. Благодаря этому организация может гибко реагировать на различные сетевые угрозы, настраивая под каждый конкретный случай необходимые ответные действия.

Перейдем теперь к вопросам о месте данного типа устройств в сетевой инфраструктуре предприятия и о его стыковке с другим оборудованием. Существует два режима использования подобного оборудования — Inline и Promiscuous. В первом случае система пропускает через себя весь трафик и может напрямую блокировать его при необходимости. Во втором случае на систему заводится зеркальный поток реального трафика, который и подвергается аналитической обработке.

Затронем проблематику внедрения подобных решений на предприятиях. Для эффективного использования подобного рода устройств необходимы квалифицированные специалисты со специфичными знаниями в области информационной безопасности, которые далеко не всегда и не везде могут быть доступны. Необходимо учесть, что однократной настройки такому устройству будет недостаточно: необходимо постоянно следить за поддержанием системы в актуальном состоянии. Также важно пристальное внимание к вопросам информационной безопасности и понимания возможных угроз и методов борьбы с ними со стороны руководства предприятия.

## Обманные системы

Обычно, когда речь заходит об обмане в области информационной безопасности, сразу вспоминаются попытки злоумышленников использовать те или иные скрытые лазейки для обхода задействуемых средств защиты. Будь то продажа паролей и работа от имени авторизованного пользователя или несанкционированное использование модемов. Однако обман может сослужить хорошую службу не только для злоумышленников, но и для защитников корпоративных ресурсов. Сразу необходимо отметить, что обман очень редко используется в качестве защитного механизма. Обычно, когда речь заходит о средствах защиты, на ум сразу приходят современнейшие межсетевые экраны, блокирующие любые попытки проникновения хакеров. Если стремиться к более мощной защите, то придется обратить свое внимание и на другие "нестандартные" защитные механизмы. Это частично собьет с толку злоумышленников и нарушителей, привыкших к широко разрекламированным средствам обеспечения информационной безопасности.

Существует множество различных вариантов использования обмана в благих намерениях. Вкратце перечислю некоторые механизмы обмана, основываясь на классификации Даннигана:

* Сокрытие
* Камуфляж
* Дезинформация

В той или иной мере эти механизмы находят применение в практике работы, систем безопасности. Однако, как правило, эти механизмы задействуются не для информационной, а для иных областей обеспечения безопасности (физической, экономической и т. д.).

В сфере защиты информации наибольшее распространение получил первый метод — сокрытие. Ярким примером использования этого метода в целях обеспечения информационной безопасности можно назвать сокрытие сетевой топологии при помощи межсетевого экрана. Примером "камуфляжа" можно также служить использование Unix-подобного графического интерфейса в системе, функционирующей под управлением операционной системы Windows NT, Если злоумышленник наткнется на такой интерфейс, то можно быть уверенным, что он будет пытаться реализовать атаки, характерные для ОС Unix, а не для ОС Windows NT. Это существенно увеличит время, необходимое для "успешной" реализации атаки. Примером дезинформации можно назвать использование заголовков (banner), которые бы давали понять злоумышленнику, что атакуемая им система уязвима. Например, если в сети используется почтовая программа sendmail версии 8.9.3, а возвращаемый ею заголовок утверждает обратное, то нарушитель потратит много времени и ресурсов, чтобы попытаться эксплуатировать уязвимости, присущие ранним версиям sendmail (до 8.9.3).

Рассмотрим только 2 и 3 классы обманных методов, как менее распростра­ненные и наиболее интересные, Работа систем, их реализующих, заключается в том, что эти системы эмулируют те или иные известные уязвимости, которых в реальности в защищаемых объектах не существует. Использование средств, реализующих камуфляж и дезинформацию (deception systems), как правило, имеет эффект наподобие описанного ниже.

* Увеличение числа выполняемых нарушителем операций и действий. По­скольку заранее определить, является ли обнаруженная нарушителем уязвимость истинной или нет, нельзя, злоумышленнику приходится тратить много дополнительных усилий, чтобы выяснить это. И даже дополнительные действия не всегда помогают в этом.
* Получение возможности отследить нападающих. За тот период времени, когда нападающие пытаются проверить все обнаруженные уязвимости, в т.ч. и фиктивные, администраторы безопасности могут проследить весь путь до нарушителя или нарушителей и предпринять соответствующие меры.

Обычно в информационной системе задействуются от 5 до 10 зарезервированных портов (с номерами от 1 до 1024). К ним можно отнести порты, отвечающие за функционирование сервисов HTTP, FTP, SMTP, NNTP, NetBIOS, Echo, Telnet и др. Если обманные системы эмулируют использование еще 100 и более портов, то работа нападающего увеличивается во сто крат. При этом мало найти открытый порт, надо еще попытаться взять под контроль уязвимости, связанные с этим портом. И даже если нападающий автоматизирует эту работу путем использования соответствующих программных средств (Nmap, SATAN и т. д.), то число выполняемых им операций все равно существенно увеличивается, что приводит к быстрому снижению производительности его работы.

Есть и другая особенность, присущая обманным системам. По умолчанию обращение ко всем неиспользуемым портам игнорируется. Тем самым попытки сканирования портов могут быть пропущены активными защитными средствами. В случае же привлечения обманных систем все эти действия будут сразу же обнаружены при первой попытке обращения к ним.

Включение в систему безопасности обманных систем — это достаточно интересный и при правильном применении — эффективный метод обеспечения информационно безопасности. Однако для большей эффективности можно порекомендовать использовать связку защитных средств "обманные системы — системы обнаружения атак", которая позволит не только обнаружить нападающего сразу же после первой попытки атаки, но и заманить его при помощи обманной системы, и тем самым дать время администраторам безопасности на обнаружение злоумышленника и принятие соответствующих мер.

Существует два класса обманных систем. Первые эмулируют некоторые сервисы или уязвимости только на том компьютере, на котором они запущены. Примером такой системы является Decoy-режим RealSecure OS Sensor, WinDog-DTK или система The Deception Toolkit (DTK). Второй класс систем эмулирует не отдельные сервисы, а сразу целые компьютеры и даже сегменты, содержащие виртуальные узлы, функционирующие под управлением разных ОС. Пример такой системы — CyberCop Sting.

Но не стоит забывать, что обманные системы — это не панацея от всех бед. Они помогают в случае простых нападений, осуществляемых начинающими или неопытными злоумышленниками. Если атаки осуществляются квалифицированными и опытными нарушителями, обманные системы теряют свою эффективность. Предварительный анализ трафика позволяет злоумышленнику понять, какие из обнаруженных портов фиктивные. Моделирование атак на стенде и сравнивание результатов с тем, что выдается в реальной атакуемой системе, также позволяет обнаружить использование обманных средств. Мало того, неправильная конфигурация обманной системы приведет к тому, чтозлоумышленник сможет обнаружить факт слежки за ним и прекратит свою несанкционированную деятельность. Однако число действительно квалифицированных злоумышленников не так велико, и поэтому использование, обманных средств может помочь в большинстве случаев.

### Примеры обманных систем

**The Deception Toolkit**

Набор инструментальных обманных средств (The Deception Toolkit, DTK) является средством для введения в заблуждение автоматизированных средств анализа защищенности путем создания ложных уязвимостей. Создаваемые сканерами безопасности отчеты не смогут помочь злоумышленнику определить, какие из обнаруженных уязвимостей являются реальными, а какие нет. Тем самым взломщику придется тратить время и ресурсы на проверку всех обнаруженных уязвимостей, что позволит своевременно обнаружить такие попытки и противопоставить им эффективные средства защиты и, возможно, обнаружить злоумышленника.

DTK может быть легко настроен под различные требования пользователей, но обычно это программирование ограничивается созданием ответных реакций на действия нападающих.

DTK представляет собой набор программ на языке С и Perl, реализующих описанные выше механизмы обмана злоумышленников. Эти программы могут быть адаптированы под конкретные нужды пользователей.

Развивая пакет DTK, автор разработал улучшенную версию — DTK-Pro, которая, сохраняя все возможности DTK, дополняет его новыми механизмами:

* централизованным управлением несколькими DTK, установленными на разных узлах сети;
* графическим интерфейсом администратора DTK-Pro;
* проверкой согласованности задаваемых обманных правил и рядом других функций.

**CyberCop Sting**

Система CyberCop Sting разработана предназначена для эмуляции Ethernet-сегмента, состоящего из маршрутизаторов Cisco с IOS 11.2 и серверов под управлением операционных систем Windows NT 4.0 и Solaris 2.6. Все посланные на смоделированные виртуальные хосты пакеты отслеживаются, что позволяет обнаруживать и отслеживать злоумышленников, в то время как они пытаются атаковать несуществующие узлы.

CyberCop Sting "создает" виртуальную сеть на выделенном узле, работающем под управлением Windows NT. Каждый из виртуальных узлов имеет один или несколько IP-адресов, на которые можно посылать сетевой трафик и получать вполне "реальный" ответ. В более сложных случаях виртуальный узел может выступать в роли ретранслятора пакетов на невидимый, но реальный компьютер, который и отвечает на все запросы злоумышленника. Главное достоинство системы CyberCop Sting в том, что для моделирования "приманки" для нарушителя вам не требуется большого количества оборудования и маршрутизаторов, — все реализуется на единственном компьютере.

Однако в CyberCop Sting отсутствует та гибкость, которая имеется в DTK: не возможно задать эмулируемые уязвимости и даже для стандартных сетевых сервисов, моделируемых Sting'oм, не возможно определить специфичные варианты ответов на запросы. Кроме того, так же, как и в DTK, CyberCop Sting не обладает качественным механизмом анализа журнала регистрации: придется просматривать сотни, а иногда и тысячи записей текста, прежде чем найдете интересующую информацию. И хотя для этой цели можно использовать продукты третьих фирм, например, WebTrends Log Analyzer, неплохо было бы иметь и встроенные средства анализа журналов регистрации.

## Системы контроля целостности

Если, несмотря на использование "классических" систем обнаружения атак и другие предпринятые защитные меры, злоумышленник все-таки проник в защищаемую систему, то, как правило, он попытается установить программы типа "троянский конь", "поправить" системные файлы или отключить систему защиты. В абсолютном большинстве случаев все эти действия сводятся к изменению каких-либо файлов (исполняемых, конфигурационных, динамических библиотек, драйверов и т. п.).

Целевой анализ (target-based) (также известный как контроль целостности файлов) опирается на пассивные, не оказывающие заметного влияния на работу контролируемой системы методы для проверки целостности системы и файлов данных, а также объектов системы и их атрибутов (например, потоков данных, баз данных и ключей системного реестра). Системы контроля целостности используют криптографические проверки контрольных сумм, для того, чтобы получить доказательства подделки для наиболее важных системных объектов и файлов. Алгоритмы этих проверок обладают таким свойством, что даже незначительные изменения во входных данных функции создают большие различия в результате. Это означает, что несущественные отклонения в потоке входных данных приведут к тому, что алгоритм контроля целостности создает значительное расхождение в контрольной сумме, генерируемой алгоритмом. Данные алгоритмы являются криптографически стойкими, т.е. при заданном конкретном входном значении (величине) практически невозможно получить идентичный результат для любого другого входного значения. Это предотвращает наиболее распространенную атаку против сравнительно простых алгоритмов генерации контрольных сумм (CRС), при которых хакеры маскируют изменения в содержании файла, так что одинаковая контрольная сумма создается как для оригинального, так и для подделанного файла.

Системы контроля целостности работают по замкнутому циклу, обрабатывая файлы, системные объекты и атрибуты системных объектов с целью получения контрольных сумм; затем они сравнивают их с результатами, полученными на предыдущем цикле, отыскивая расхождения. Когда различие обнаружено, продукт посылает сообщение администратору безопасности, при этом фиксируя время, соответствующее времени вероятного изменения.

Контроль целостности позволяет реализовать стратегию эффективного мониторинга, сфокусированную на системах, в которых целостность данных и целостность процессов играет наиболее важную роль (например, системах управления базами данных). Этот подход дает возможность контролировать конкретные файлы, системные объекты и атрибуты системных объектов на происходящие изменения, обращая особое внимание скорее на ***конечный*** результат атаки, **а** не на ***подробности*** развития атаки. Подробная информация о преимуществах и недостатках систем контроля целостности содержит­ся в табл.

***Таблица*** *Достоинства и недостатки систем контроля целостности*

|  |  |
| --- | --- |
| **Достоинства** | **Недостатки** |
|  Любая успешная атака, при которой были изменены файлы, даже если использовались rootkits или перехватчики сетевых пакетов, будет определяться независимо от того, задействован ли для определения атаки анализ сигна­тур или статистический анализ. Поскольку нет зависимости от старых записей режимов работы, контроль целостности может обнаруживать атаки, которые другим методоло­гиям выявить не по силам. Этот подход допускает надежное обнаружение как местоположения, так и наличия атак, которые видоизменяют систему (например, "троянских коней"). Из-за того, что собственные воздействия и влияния данного механизма являются незначительными, контроль целостности может быть полезным для мониторинга систем с умеренной поло­сой пропускания для обработки данных. Средства контроля целостности являются эффективными для определения того, какие файлы необходимо заменить с целью восстановления системы вместо переинсталляции всегда содержимого с оригинального источника или с резервной копии, как это часто делается |  Поскольку современные реализа­ции этого подхода стремятся работать в пакетном (batch) - режиме, они приводят к реагированию на атаки не в реальном масштабе времени. В зависимости от количества файлов, системных объектов и атрибутов объектов, для которых вычисляются контрольные суммы, рассматриваемые средства могут все же оказать заметное влияние на производительность системы. Эти средства не очень хорошо подходят для осуществления об­наружения в реальном масштабе времени, поскольку они контролируют результаты атак, а не сами атаки в процессе их развития |

Существует очень большое число систем контроля целостности, как распространяемых на коммерческой основе, так и условно-бесплатных. В качестве примера приведу только несколько таких систем — Tripwire, L5, AIDE.

### Примеры систем контроля целостности

**Tripwire**

Изначально система Tripwire была разработана одним из американских университетов и являлась условно бесплатной.

Однако позже эта система стала развиваться компанией Tripwire Security и в настоящий момент представляет собой коммерческий продукт. Имеется несколько основных отличий данной системы от своих условно-бесплатных собратьев. Первое и основное отличие — поддержка архитектуры "клиент-сервер". Если все остальные системы работают в автономном режиме, Tripwire состоит из двух основных компонентов: HQ Connector и HQ Сonsole. Первый компонент устанавливается на удаленные узлы, работающие под управлением различных ОС, и осуществляет контроль целостности данных объектов. Второй компонент управляет удаленными агентами (НQConnector) и выполняет централизованный сбор данных от них, генерации отчетов и другие функции. Вторая особенность Tripwire, отсутствующая, у других аналогичных систем — поддержка платформы . Tripwire поддерживает следующие ОС: Windows NT, Solaris, Linux, AIX, HP UX, IRIX и Tru64. Итретье отличительное свойство — наличие языка описания политики контроля целостности (Enhanced Policy Language).

Компания Tripwire Security выпустила в рамках проекта GNU свободно распространяемую версию системы Tripwire 3 ASR, которая отличается от коммерческого релиза следующими возможностями:

* отсутствием поддержки Windows NT;
* исключением уведомления об изменении контрольных сумм по электронной почте;
* меньшим числом контролируемых атрибутов (9 против 14);
* "слабой" системой генерации отчетов (нет сортировки и детализации);
* отсутствием централизованной графической консоли управления;
* изъятием возможности назначения приоритетов проверок и защиты своих файлов от НСД.

**IDE и L5**

По мнению некоторых специалистов, система AIDE (Advanced Intrusion Detection Environment), доступная по адресу **http://www.cs.tut.fi/~rammer/Ie.HTML** и L5 (ftp://avian.org/src/hacks/) являются лучшими альтернативами свободно распространяемой версии системы Tripwire.

**Gog&Magog**

В отличие от рассмотренных выше средств, система Gog&Magog (**tp://www.multimania.coin/cparisel/gog/)** не только генерирует и проверяет контрольную сумму (по протоколу MD5) для выбранных файлов (в автома­тическом режиме), но и контролирует другие параметры файлов (права дос­тупа, владелец и т.д.).

Различные категории систем обнаружения атак имеют свои достоинства и преимущества, а также свои слабые стороны. Совместное использование разноплановых средств позволит усилить достоинства одних и устранить недостатки других, тем самым помогая выстроить эффективную инфра­структуру обнаружения атак. Комбинирование всех этих технологий значи­тельно улучшит стойкость корпоративной сети к атакам и злоупотреблени­ям, позволит ужесточить политику безопасности и внести большую гибкость в процесс эксплуатации сетевых ресурсов.