СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

Р.Р. Фаткиева

Основы построения защищенных компьютерных сетей

Протокол TCP

СПбГЭТУ «ЛЭТИ», 2021 г.

3.6 Протокол TCP

В стеке протоколов TCP/IP протокол TCP (Transmission Control Protocol) работает на транспортном уровне, обеспечивая надежную транспортировку данных между прикладными процессами путем установления логического соединения.

К функциям TCP относят:

- получение и передача потока данных от вышестоящего уровня IP-модулю;
- обеспечение полнодуплексной передачи данных;
- обеспечение защиты от повреждения, потери, дублирования;
- обеспечение работы нескольких соединений;
- управление потоком данных с помощью механизмов скользящего окна).

Формат сообщений TCP. Единицей данных протокола TCP является сегмент. Информация, поступающая к протоколу TCP в рамках логического соединения от протоколов более высокого уровня, рассматривается протоколом TCP как неструктурированный поток байт. Поступающие данные буферизуются средствами TCP. Для передачи на сетевой уровень из буфера "вырезается" некоторая непрерывная часть данных, называемая сегментом, состоящая из заголовка и блока данных. Заголовок сегмента имеет следующие поля (рис. 3.7).

Порт источника (Source Port) и *порт назначения* (Destination Port) занимают 16 бит. Протокол TCP обеспечивает работу одновременно несколько соединений. Каждый прикладной процесс идентифицируется IP-адресом и номером порта.

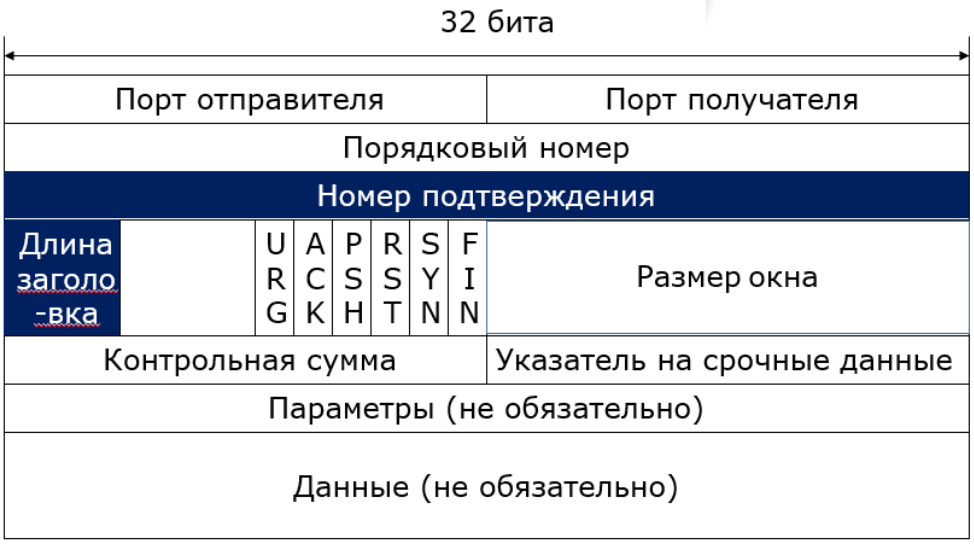

Рис. 3.7

Порядковый номер (Sequence number) занимает 32 бита, указывает номер байта, который определяет смещение сегмента относительно потока отправляемых данных;

Подтвержденный номер (Acknowledgment number) занимает 32 бита, содержит максимальный номер байта в полученном сегменте, увеличенный на единицу; т.е все предыдущие байты были получены именно это значение используется в качестве квитанции;

Длина заголовка (Data Offset) занимает 4 бита, указывает длину заголовка сегмента TCP, измеренную в 32-битовых словах. Длина заголовка не фиксирована и может изменяться в зависимости от значений, устанавливаемых в поле Опции. Служит указателем на начало пол данных;

Резерв (Reserved) занимает 6 битов, поле зарезервировано для последующего использования. Заполняется нулями;

Кодовые биты (Control bits) занимают 6 битов, содержат служебную информацию о типе данного сегмента, задаваемую установкой в единицу соответствующих бит этого поля:

URG - срочное сообщение, используется если приложение обращается с запросом о срочной передаче данных. В этом случае протокол TCP, не ожидая заполнения буфера до уровня размера сегмента, немедленно передает указанные данные в сеть. Полученные из сети данные минуя буфер

передаются процессу или приложению. Данная технология используется для передачи видео и аудиоданных;

ACK - квитанция на принятый сегмент;

PSH - запрос на отправку сообщения без ожидания заполнения буфера;

RST – сброс текущего соединения (при получении соединение ликвидируется, недопоставленные данные уничтожаются);

SYN - запрос на установление соединения;

FIN - признак достижения передающей стороной последнего байта в потоке передаваемых данных.

Окно (Windows) занимает 16 бит, содержит объявляемое значение размера окна в байтах;

Контрольная сумма (Checksum) занимает 16 бит, определяется для блока данных, состоящего из псевдозаголовка и самого сегмента данных. 96 -битный псевдозаголовок предшествует заголовку TCP и содержит IP-адрес отправителя, получателя, идентификатор протокола и сегмента.

Указатель срочности (Urgent Pointer) занимает 16 бит, используется совместно с кодовым битом URG и указывает на конец данных, которые необходимо срочно принять, несмотря на переполнение буфера;

Опции (Options) — это поле имеет переменную длину и может вообще отсутствовать, используется для решения вспомогательных задач, например, при выборе максимального размера сегмента, или передачи дайджеста MD5;

Заполнитель (Padding) может иметь переменную длину, представляет собой фиктивное поле, используемое для доведения размера заголовка до целого числа 32-битовых слов.

Порты и установление TCP-соединений. Для организации надежной передачи данных предусматривается установление логического соединения между двумя прикладными процессами. В рамках соединения осуществляется обязательное подтверждение правильности приема для всех переданных сообщений, и при необходимости выполняется повторная передача. Соединение в TCP позволяет вести передачу данных одновременно в обе стороны, то есть полнодуплексную передачу (рис 3.8).

Рис. 3.8

Соединение в протоколе TCP идентифицируется парой полных адресов обоих взаимодействующих процессов, включающих IP-адрес (номер сети и номер компьютера) и номер порта. Портам присваиваются стандартные, зарезервированные номера (например, номер 21 закреплен за сервисом FTP, 23 - за TELNET), или произвольно выбранными локальными номерами.

Установление соединения выполняется в следующей последовательности (рис. 3.8):

1. Узел А посылает узлу В запрос на открытие порта, а также запрос процессу, с которым требуется установить соединение (active open).

2. Узел В открывает порт для приема данных (passive open) и возвращает квитанцию, подтверждающую прием запроса.

3. Получив квитанцию, узел А открывает порт для передачи (active port) и передает запрос к противоположной стороне.

4. Соединение считается установленным. Далее происходит обмен данными в рамках данного соединения.

Предположим, что узел А устанавливает соединение с узлом В. Для этого:

1. Узел А в сообщении TCP, посылаемому узлу В устанавливает флаг SYN и начальный порядковый номер ISN с которого будут нумероваться отправляемые данные.

2. Для подтверждения приема сообщения, узел B откликается посылкой TCP сегмента с установленным флагом ACK. Но вследствие того, что протокол TCP обеспечивает полнодуплексную передачу, узел B может в свою очередь запросить соединение на передачу данных с узлом A посылкой флага SYN и начального порядкового номера своих сообщений ISN.
3. Узел A, подтверждает получение сообщение от узла B. Так как сеанс связи можно считать установившимся, то узел A может включить свои данные, нумерация которых начинается с ISN (A)+1 в это сообщения.
4. Происходит передача данных между узлами A и B.
5. Сеанс обмена данными заканчивается процедурой закрытия, которая в заголовке последнего сегмента использует флаг FIN. Аварийный разрыв соединения происходит посылкой сообщение с битом RST, при этом все недопоставленные данные уничтожаются.

Концепция квитирования. В рамках соединения правильность передачи каждого сегмента должна подтверждаться квитанцией получателя. Квитирование — это один из традиционных методов обеспечения надежной связи. Идея квитирования состоит в следующем.

Для организации повторной передачи искаженных данных отправитель нумерует отправляемые кадры и ожидает от приемника *положительную квитанцию* - служебное сообщение, извещающее о том, что исходный кадр был получен и данные в нем оказались корректными. Существуют два подхода к организации процесса обмена квитанциями: с простоями и с организацией "окна".

Метод с простоями требует, чтобы узел, посылал очередной кадр, после получения квитанции (положительной или отрицательной) от получателя. Производительность обмена данными в этом методе незначительна, так как передающий узел и мог бы послать следующий кадр сразу же после отправки предыдущего, однако он обязан ждать прихода квитанции, что особенно заметно на низкоскоростных каналах связи, то есть в территориальных сетях.

В *методе с организацией "окна"* отправителю разрешается передать некоторое количество кадров в непрерывном режиме, без получения на эти кадры ответных квитанций. Количество кадров, которые возможно передать таким образом, называется размером окна W. При отправке кадра с номером 1 отправителю разрешается передать еще (W-1) кадров до получения квитанции на первый кадр. Если квитанция на кадр 1 не получена, то процесс передачи приостанавливается, и по истечению некоторого тайм-аута кадр 1 считается утерянным и передача его осуществляется снова. Выбор времени ожидания (тайм-аута) очередной квитанции является важной задачей, при которой необходимо учитывать скорость и надежность линий связи, их протяженность и т.д. В протоколе TCP при каждой передаче засекается время от момента отправки сегмента до прихода квитанции о его приеме. Получаемые значения усредняются с весовыми коэффициентами, возрастающими от предыдущего замера к последующему. В качестве тайм-аута выбирается среднее время оборота, умноженное на некоторый коэффициент.

Описанный алгоритм называется *алгоритмом скользящего окна* (при каждом получении квитанции окно перемещается (скользит), захватывая новые данные, которые разрешается передавать без подтверждения).

В протоколе TCP квитанцией (ASK SN) подтверждается правильный прием данных, отсутствие квитанции говорит о приеме искаженного сегмента, потере сегмента или квитанции. В качестве квитанции получатель отсылает сообщение (сегмент), в которое помещает число, на единицу превышающее максимальный номер байта в полученном сегменте. Если размер окна равен W, а последняя квитанция содержала значение n, то отправитель может посылать новые сегменты до тех пор, пока в очередной сегмент не попадет байт с номером n+W. Этот сегмент выходит за рамки окна, и передачу в таком случае необходимо приостановить до прихода следующей квитанции.

Изменяя величину окна, возможно, повлиять на загрузку сети. Чем больше окно, тем большую порцию неподтвержденных данных можно послать в сеть. Если сеть не справляется с нагрузкой, то протокол TCP, отправляя квитанцию, помещает в нее новый, уменьшенный размер окна. Если узел

совсем отказывается от приема, то в квитанции указывается окно нулевого размера. После приема квитанции с нулевым значением окна отправитель время от времени делает контрольные попытки продолжить обмен данными. Если протокол-приемник уже готов принимать информацию, то в ответ на контрольный запрос он посылает квитанцию с указанием ненулевого размера окна.

3.7 Обеспечение безопасности протокола TCP/IP

Основным недостатком протокола TCP/IP является отсутствие встроенных функций защиты, криптографического закрытия пакетов сообщений, контроля подлинности, а также аутентификация участников обмена. Поэтому пакеты сообщений доступны для анализа и модификации.

В случае пассивных атак на протоколы TCP/IP, возникает трудность обнаружения нарушителя, все несанкционированные действия которого сводятся к наблюдению за циркулирующими в сети пакетами. При активных атаках на TCP/IP нарушитель модифицирует и/или фильтрует содержимое пакетов сообщений для обмана получателя или нарушения работоспособности принимающего узла.

Действия нарушителя, направленные против какого-либо узла или сети, представленные в таблице 2:

- несанкционированное подключение и прослушивание сети, с последующим анализом потока информации;
- прерывание передачи сообщений;
- перехват, модификация данных, передаваемых через сеть;
- выдача себя за другой узел, для использования его привилегий;
- несанкционированная передача данных (обход правил фильтрации IP трафика в сетях);
- отказ в обслуживании- приведение узла в состояние, когда он не может нормально функционировать, принимать и передавать данные.

Таблица 2.

Реализация атаки	Противодействие
Прослушивание. Осуществляется переводом интерфейса персонального компьютера в режим прослушивания, при этом возможен отбор трафика	1.Разбиение сети на сегменты с помощью коммутаторов, при котором злоумышленник

по установленным критериям (tcpdump, noop). Наиболее подвержены сети Ethernet, FDDI, радиосети, из- за сложности физического доступа несколько сложнее реализация для выделенных телефонных и коммутируемых линий.	перехватывает данные, получаемые или отправляемые узлами сегмента, к которому он подключен. 2.Шифрование передаваемых данных. 3. Обнаружение злоумышленника с помощью специальных программ (AntiSniff): - Распространяя по сети сообщение ICMP Echo внутри «неверного» кадра, который при нормальных обстоятельствах должен быть проигнорирован, и получив на него ответ. - Распространяя по сети дейтаграммы не предназначенные ни одному из узлов тестируемой сети и прослушивая DNS запросы от узлов о доменных именах для этих фиктивных адресов, находящимися в режиме прослушивания. - Распространяя по сети сообщение ICMP Echo и фиксируя время отклика, порождая затем шквал кадров на несуществующие MAC-адреса, при этом измеряя время отклика, которое у узлов переведенных в режим прослушивания многократно увеличивается.
Сканирование сети - выявление подключенных к сети компьютеров и определение работающих на них сетевых сервисов 1.1 Сканирование подключенных к сети компьютеров осуществляется с помощью программы ping посылкой ICMP-сообщений Echo по широковещательному адресу, на которое ответят все компьютеры, поддерживающие обработку данных сообщений. 1.2. Обратное сканирование реализуется посылкой RST-сегментов, ответы на несуществующие DNS-запросы и т.д. Если ответное сообщение не	1. Анализ трафика и отслеживания Echo-сообщений, посылаемых последовательно по всем адреса сети. 2. Анализ системного журнала и обнаружение многочисленных и сразу же прерванных соединений. 3. Запрет пропускания брандмауэрами SYN пакетов без флага ACK во внутреннюю сеть из Интернета. Запрет соединений инициируемые хостами из

получено, то узел существует, подключен к сети и работает. 2.1 Выявление подключенных компьютеров позволяет определить помощью программ сканеров открытые порты, для реализации атак на уровне приложений. Например, с помощью стандартной программы telnet возможна попытка установления TCP соединения с помощью функции connect, позволяющей указать произвольный номер порта для установления соединения. 2.2 Сканирование в режиме половинного открытия. Для этого на сканируемый порт направляется SYN-сегмент, получение ответного сегмента с битами SYN и ACK означает, что порт открыт, сегмент с битом RST –порт закрыт. В случае если попытка открытия порта осуществилась, то злоумышленник немедленно ликвидирует попытку соединения отправляя сегмент с битом RST. 2.3 Сканирование с помощью FIN сегментов. На сканируемый порт посылается пакет с установленным битом FIN (PSH,URG), получение ответного сегмента с битами RST означает, что порт закрыт, в противном случае порт открыт.	Интернета с внутренними хостами. 4. Обнаружение злоумышленника с помощью специальных программ (tcplogd)

Перехват данных. 1. Ложные ARP-ответы. Используется для перехвата трафика между узлами A и B, с помощью рассылки сфальсифицированных ARP-сообщений, при котором каждый из атакуемых узлов считает MAC-адрес злоумышленника адресом собеседника. Для этого: 1. С помощью программы ping злоумышленник X определяет MAC адреса узлов A и B. 2. На компьютере злоумышленника конфигурируется дополнительные логические IP интерфейсы, с именами адресов A и B, и отключенным для них протоколом ARP. Устанавливается статическая ARP-таблица и разрешается ретрансляция дейтаграмм. 3. На MAC-адреса узла A отправляется сфабрикованный ARP-ответ, в котором сообщается, что IP-адресу B соответствует MAC – адрес X. Аналогичный ответ отправляется узлу B. В связи с тем, что ARP-протокол без сохранения состояния, то узлы A и B примут ARP-ответ, даже если они не посылали запрос. 4. Для поддержания в «неведении» узлы с интервалом 20-40 с. посылаются ARP-ответы, для обновления сфабрикованных записей в ARP-таблицах узлов A и B. В случае если узел B является шлюзом сети, в котором находится узел A, то возможен перехват всего трафика между узлом A и Интернетом. Злоумышленник X может выступать в роли узла B, фабрикуя ответы от его имени, и отсылая их в A.	1. Введение базы данных соответствий MAC адресов и IP адресов всех узлов сети. 2. Использование специальных программ, прослушивающих сеть и уведомляющих о нарушениях (arpwatch)/ 3. Использование статических ARP- таблиц.
Навязывание ложного маршрутизатора ***Ложное сообщение ICMP Redirect*** Навязывание узлу A адреса X, в качестве адреса маршрутизатора. При этом данные от узла A перехватываются X, анализируются и отправляются настоящему маршрутизатору, который пересылает их узлу B. Для этого: 1. X должен находится в одной IP-сети с узлом A и знать адрес маршрутизатора, через который A отправляет дейтограммы B. 2. X формируем IP-дейтограмму, в которой в качестве отправителя указан IP адрес шлюза, а получателя IP адрес узла A. В дейтограмму помещается сообщение Redirect, где в поле адрес нового маршрутизатора указан IP- адрес X. Сформированное сообщение отправляется A.	1. Отключение в узле обработки сообщений Redirect. 2. Использование специальных программ, выявляющих дополнительные промежуточные узлы (traceroute).

3. Узел А получив дейтограмму, считает, что Redirect является реакцией на ранее отправленную дейтограмму узлу В. Из возвращенного заголовка и первых 64 бита узел А определяет, что необходимо сделать перенаправление для дейтограмм и вносит в свою таблицу маршрутов частный маршрут через маршрутизатор, указанный в сообщении Redirect. 4. Новый маршрут действует несколько минут, поэтому Х необходимо периодически повторять действия 1-3.	
Атака на протоколы маршрутизации. Злоумышленник может попытаться ввести в заблуждение маршрутизаторы. Для этого формируются подложные сообщения протокола маршрутизации с целью переключения требуемых маршрутов на себя. 1. Злоумышленник Х расположен между двумя сетями А и В. 2. Перехватив сообщения RIP- между сетями А и В, злоумышленник отправляет сфальсифицированное сообщение узлу А, где в векторе расстояния указывает расстояние от себя до узла А. То же производится с узлом В. Тем самым заставляя маршрутизаторы изменять свою таблицу маршрутизации. 3. В случае, если расстояние от маршрутизатора А к В =1, то злоумышленник формирует маршрутизатору А сообщение от имени В с вектором =16, заставляя тем самым удалить данный маршрут из своей таблицы. Затем злоумышленник формирует новое сообщение от своего имени вектором=1, с указанием доступности сети В через Х.	1. Запрет размещения хостов в транзитивных сетях. 2. Аутентификация сообщений протокола маршрутизации с помощью алгоритма MD5 3. Блокирование обновлений приходящих от сетей, которые считаются опасными.
Несанкционированный обмен данными **Туннелирование**. Злоумышленник хочет отправить данные с узла Х узлу А, однако правила фильтрации запрещают отправку дейтограмм узлу А, но разрешают узлу В. Для осуществления данной атаки злоумышленник должен: 1. Для этого злоумышленником формируется дейтограмма, заголовок которой заполняется стандартным способом. Данная дейтограмма инкапсулируется в новую IP-дейтограмму, при этом в поле протокола указывается, инкапсуляция и адрес промежуточного узла В. 2. Получив данную дейтограмму узел В анализирует первый заголовок видит, что	1. Запрет пропуска фильтрующим узлом дейтограмм с опциями Lcose/strict 2. Фильтрация трафика по адресу отправителя 3. Запрет трансляции дейтограмм с полем Protocol=4

дейтограмма адресована не узлу А и отправляет ее по назначению, снабдив новым заголовком 3. В данном случае фильтрующий узел видит, что дейтограмма пришла с разрешенного узла и пропускает ее. **Атака крошечными фрагментами** 1. Злоумышленник формирует датаграмму TCP-сегмента, состоящую из 2 фрагментов: в первом фрагменте датаграммы равной 8 октетам, находятся номера портов отправителя и получателя и поле Sequence Number, однако значения флагов не попадает. 2. Маршрутизатор, используя правила фильтрации проверяет первый фрагмент датаграммы, на наличие флага SYN, так как он отсутствует, то дейтограмма пропускается в сеть. 3. Злоумышленник формирует 2 фрагмент дейтограммы, помещая оставшуюся часть заголовка TCP- сегмента, с установленным флагом SYN 4. Так как первый фрагмент дейтограммы проверен, остальные фрагменты маршрутизатором проверяться не будут, и дайтограмма будет пропущена в сеть. 5. Происходит сборка дейтограммы на узле получателе и соединение с компьютером злоумышленника **Атака накладывающими фрагментам.** 1. Злоумышленник формирует датаграмму TCP-сегмента, состоящую из 2 фрагментов: в первом фрагменте равном 8 октетам находится полный TCP-заголовок, с установленным флагом ACK. 2. Маршрутизатор, используя правила фильтрации проверяет первый фрагмент датаграммы, на наличие флага SYN, так как он отсутствует, то дейтограмма пропускается в сеть. 3. Злоумышленник формирует 2 фрагмент дейтограммы, начиная с 9 октета с установленным флагом SYN 4. Так как первый фрагмент дейтограммы проверен, остальные фрагменты маршрутизатором проверяться не будут, и дайтограмма будет пропущена в сеть. 5. Происходит сборка дейтограммы на узле получателе. При этом второй фрагмент накладывается на первый. При сборке такой датаграммы в TCP-заголовке переписываются поля начиная с ACK SN в соответствии со значениями из	Запрет пропуска маршрутизатором датаграммы с Fragment Offset=0, 1 и Protocol=6 (TCP), размер поля данных которых меньше 8 октетов.

второго фрагмента, и в итоге получается SYN-сегмент.	
Имперсонализация (spoofing) **Десинхронизация**. Злоумышленник X находится в одном сегменте сети с узлами A и B или на пути между ними. Каждый из узлов A и B при установлении связи знает номера дейтограмм, которые может прислать собеседник, а также размер окна. Если дейтограмма не попадает в окно, то она уничтожается, а собеседнику отправляется сообщение с номером ожидаемой дейтограммы. Для реализации атаки: 1. Злоумышленник сбивает показания счетчиков узлов A и B. 2. Если узел B отправляет узлу A дейтограмму с номером SN , не попадающем в окно A ,то узел A уничтожает дейтограмму и отправляет сообщение с указанием ожидаемого SN. Но данный SN не попадает в окно B. (ACK-шторм) 3. В данном случае, злоумышленник выступает в качестве посредника и может не только подслушивать, но активно подменять трафик.	1. Запрет пропуска маршрутизатором дейтограмм: ○ приходящих на внешний интерфейс, но имеющих адрес отправителя из внутренней сети; ○ приходящих на внутренний интерфейс, но имеющих адрес отправителя из внутренней сети. 2. Шифрование трафика и аутентификация
Отказ в обслуживании **Dos** Приведение атакуемого узла или сети в такое состояние, когда передача данных другому узлу становиться невозможной Smurf От имени атакуемого узла производится рассылка широковещательных Echo-сообщений. Если число возвращений велико, то атакуемый узел и сети-усилители могут быть заблокированы шквалом ответных сообщений **SYN-flood** Состоит в посылке злоумышленником SYN-запросов на атакуемый узел в количестве большем, чем тот может обработать. Для этого: 1. Злоумышленник от имени несуществующего отправителя посылает SYN–запросы атакуемому узлу с определенной частотой. 2. При получении SYN-запроса модуль TCP создает блок обслуживания и отправляет подтверждение в виде SYN, на которое он никогда не получит ответ. В конце концов ресурсы узла истощаются и происходит отказ.	1. Запрет на маршрутизацию дейтограмм с широковещательным адресом назначения между сетью организации и Internet. 2. Запрет на обработку узлами Echo-запросов направленных на широковещательный адрес. 3. Запрет на маршрутизацию дейтограмм, направленных и внутренней сети в Internet, но имеющих внешний адрес отправителя. 1. Использование ПО, позволяющее установить максимальное число открываемых соединений и списка разрешенных клиентов. 2. Использование маршрутизатора или межсетевого экрана, как посредника (При получении SYN–запроса маршрутизатор не ретранслирует его во внутреннюю сеть, а сам отвечает на запрос, если

	соединение с клиентом устанавливается, то маршрутизатор устанавливает соединение от имени клиента и выступает в качестве посредника)

3.8 Протокол доставки пользовательских дейтаграмм UDP

UDP (User Datagram Protocol)–протокол транспортного уровня, без гарантий доставки. Протокол UDP используется следующими прикладными процессами NFS (Network File System- сетевая файловая система), TFTP (Trivial File Transfer Protocol-простой протокол передачи файлов), SNMP (Simple network Management Protocol-простой протокол управления сетью) DNS (Domain Name Service-доменная служба имен).

Формат сообщений UDP. Единица данных протокола UDP называется UDP-пакетом или пользовательской дейтаграммой (user datagram), состоит из заголовка и поля данных, в котором размещается пакет прикладного уровня. Заголовок имеет простой формат и состоит из четырех двухбайтовых полей:

0 7	15	23	31
Source Port		Destination Port	
Message length		Checksum	

- Source Port - номер порта процесса-отправителя.
- Destination Port - номер порта процесса-получателя.
- Message length - длина UDP-пакета в байтах.
- Checksum - контрольная сумма UDP-пакета. Контрольная сумма вычисляется, как и в заголовке TCP.

При передаче сообщения пользовательские данные, поступившие от прикладного уровня, снабжаются UDP-заголовком, и отправляются на уровень IP. Данные, поступившие от прикладного уровня, рассматривается как целостное сообщение и никогда не разбивается на несколько пакетов. При необходимости передачи n сообщений, модулем UDP будет сформировано и отправлено n пакетов.

При получении сообщения проверяется контрольная сумма, если она соответствует, то сообщение передается прикладному процессу на порт,

указанный в поле Destination Port. В случае, если контрольная сумма не соответствует, порт не открыт, пакеты не успевают обрабатываться, то обобщение игнорируется, а отправителю возможна отправка сообщение об ошибке.

1.1 Литература

1. Таненбаум, Эндрю. Компьютерные сети [Текст] : учебное пособие / Э. Таненбаум; [Пер. с англ. В. Шрага], 2003. 991 с. 60
2. Олифер, Виктор Григорьевич. Компьютерные сети. Принципы, технологии, протоколы [Текст] : учеб. пособие для вузов по направлению ”Инфор матика и вычислит. техника” и по специальности ” Вычислит. машины, комплексы, системы и сети”, ”Автоматизир. машины, комплексы, системы и сети”, ” Програм. обеспечение вычислит. техники и автоматизир. систем” / В.Г. Олифер, Н.А. Олифер, 2006. 957 с. 133
3. Борисенко, Константин Алексеевич. Сети и телекоммуникации [Текст] / К. А. Борисенко, 2021. 43 с. 50 4 Дибров, Максим Владимирович. Сети и телекоммуникации. Маршрутизация в IPсетях в 2 ч. Часть 2 [Электронный ресурс] : Учебник и практикум для вузов / Дибров М. В., 2021. 351 с неогр.
4. Дибров, Максим Владимирович. Сети и телекоммуникации. Маршрутизация в IPсетях в 2 ч. Часть 1 [Электронный ресурс] : Учебник и практикум для вузов / Дибров М. В., 2021. 333 с неогр.
5. Шелухин О. И. Обнаружение вторжений в компьютерные сети (сетевые аномалии). Учебное пособие для вузов [Электронный ресурс] / О. И. Шелухин, Д. Ж. Сакалема, А. С. Филинова, 2018. 220 с