Информационная безопасность систем и технологий

Максим Захаров

Содержание

1. Протокол IP

Протокол сетевого уровня, обеспечивает передачу дейтаграмм между компьютерами, разбиение их на фрагменты и её маршрутизацию.

Протокол IP не гарантирует надёжную доставку дейтаграмм, он не управляет потоком данных и не выявляет ошибки передачи. Оптимизация маршрута проводится только для соседних узлов.

Максимальный размер пакета равен 2^{16} байт.

Связь по протоколу поддерживается без установления соединения.

Протокол IP использует 32-разрядные адреса компьютера в версии 4. В версии 6 используются 128-разрядные адреса.

1.1. Заголовок протокола IP

Длина заголовка равна 5 или 6 32-разрядных слов.
Поля:

1. *Версия*. В нём содержится версия протокола IP. Это поле указывает программе-получателю как декодировать все остальные поля заголовка, т. к в разных версиях поля отличаются. Если программа не работает с указанной версией протокола, то дейтаграмма отбрасывается.
2. *Длина заголовка*. В нём хранится длина заголовка. Размер поля 4 бита. В нём написано либо число 5, либо 6.
3. *Тип обслуживания*. Длина 8 бит. Первые 3 бита обозначают приоритет дейтаграммы. Если он равен 0 — обычная, все 1 — сетевое управление. 1 бит — задержка, следующий — пропускная способность, далее — надёжность.
4. *Длина дейтаграммы*. Длина поля 16 бит.
5. *Идентификатор*. Длина 16 бит. В нём содержится уникальный идентификатор, присвоенный дейтаграмме передающим узлом. Идентификатор получают от протоколов вышестоящего уровня. Он используется для правильной сборки сообщений при фрагментации.
6. *Флаги*. 3 бита. 1 бит не используется, 2 бит — флаг не фрагментировать, 3 бит — есть ещё фрагменты.
7. *Смещение фрагмента*. Номер фрагмента в сообщении. Его длина 13 бит. Смещение указывается в единицах, кратных 8 байтам.

8. *Время жизни*. Длина 8 бит. Время в секундах, которое отводится на доставку дейтаграммы. Обычно 15-30 секунд. Если время жизни истекло, дейтаграмма уничтожается. Каждый узел, в который попадает дейтаграмма, уменьшает время жизни на 1 или больше.
9. *Протокол*. Длина 8 бит. В нём содержится код протокола транспортного уровня, для которого предназначена дейтаграмма. Для TCP — 6.
10. *Контрольная сумма заголовка*. Длина 16 бит. Контрольная сумма вычисляется по всему заголовку и меняется в каждом узле, через которую она проходит. Она вычисляется путём сложения всех 16-разрядных слов заголовка и дополнением результата 1.
11. *IP-адрес отправителя*. 32 бита.
12. *IP-адрес получателя*. 32 бита.
13. *Опции*. Каждой опции отводится 1 байт. Каждый байт делится на 3 части.

    - 1 часть — копия, занимает 1 бит (нужно ли копировать поле “опции” при фрагментации дейтаграмм).
    - 2 часть — класс опции, занимает 2 бита (класс 00 — управление сетью, класс 10 — отладка сети).
    - 3 часть — номер опции, 5 бит (00000 — конец списка опций, 00011 — совместная маршрутизация, где часть маршрута задаётся отправителем, остальные определяются шлюзами, 01001 — маршрутизация отправителя, где все узлы задаются отправителем, 00111 — обозначает запись маршрута, 00100 — запись временных меток).

14. *Заполнитель*. Для дополнения заголовка до целого числа 32-разрядных слов.

После формирования дейтаграммы конструируются заголовок. Для него вычисляется контрольная сумма. Определяется узел, в которую предполагается отправить дейтаграмму. Если получатель находится в той же сети, что и отравитель, этим следующим узлом будет получатель. Если нет, то следующим узлом будет шлюз. Если используется опция “запись маршрута”, то добавляются адреса узлов нужного маршрута. Дейтаграмма передаётся через протоколы нижележащих уровней в сеть.

Каждый шлюз в сети, через который проходит дейтаграмма проверяет контрольную сумму. Если контрольные суммы не совпадают, дейтаграмма уничтожается, а отправителю передаётся сообщение об ошибке. Если совпадают, уменьшается значение поля “время жизни”, если это поле равно 0, то дейтаграмма уничтожается.

Определяется следующий узел маршрута, исходя из адреса получателя или опции маршрутизации. Заголовок перестраивается заново. Если при этом требуется фрагментация и соответствующий флаг не фрагментируется, дейтаграмма уничтожается и отправителю отправляется сообщение об ошибке.

При необходимости может быть записана временная метка. После того, как дейтаграмма достигла машины получателя, проверяется контрольная сумма заголовка. Получатель ждёт сборки в течение определённого времени. Если за это время сообщение не было собрано, то все его полученные фрагменты уничтожаются, отправителю отправляется сообщение об ошибке. Если всё нормально, IP заголовков уничтожаются, сообщение передаётся на более высокий уровень, если требуется, отправителю посылается ответ.

2. Протокол ICMP (Internet control message protocol)

Этот протокол сообщает отправителю об ошибках в сети. Используется совместно с IP. ICMP-дейтаграмма снабжена IP-заголовком, поэтому она в сети обрабатывается также, как обычная IP-дейтаграмма. В узлах сети ICMP-дейтаграммы обрабатываются на сетевом уровне.

Сообщения об ошибках передаются машине отправителя, а внутри ICMP-сообщение. Внутри этого ICMP-сообщения находятся IP-заголовок и первые 64 бита дейтаграммы, при передаче которой возникла ошибка.

Вид заголовка хранится в ICMP-заголовке. Заголовок состоит из 3 полей:

1. Тип сообщения.

   - 0 — эхо-ответ;
   - 8 — эхо-запрос;
   - 3 — адресат недостижим. Это же сообщение генерируется, если шлюзу необходимо фрагментировать дейтаграмму, а в ней установлен флаг “не фрагментировать”;
   - 4 — снизить скорость передачи данных;
   - 5 — переадресовать. Служебное сообщение для шлюзов при выполнении маршрутизации;
   - 11 — время жизни дейтаграммы истекло;
   - 12 — неправильный параметр. Возникает при обнаружении семантической или синтаксической ошибки в IP-заголовке;
   - 13 — запрос временной метки;
   - 14 — отклик на запрос временной метки. Они нужны для контроля прохождения дейтаграмм через узлы сети. При этом внутри ICMP-сообщения в запросе записывается вместо IP-заголовка исходная временная метка, а в отклике к этой метке добавляется метка получения запроса шлюза и метка отправки ответа шлюзу;
   - 17 — запрос адресной маски;
   - 18 — отклик на запрос адресной маски. Эти сообщения применяются для тестирования определённой подсети с заданной маски.

2. Код сообщения.

3. Контрольная сумма ICMP-заголовка. Вычисляется также, как контрольная сумма IP-заголовка.

3. Протокол IP версии 6

Основное отличие от версии 4 заключается в использовании 128-битных IP-адресов. Кроме этого протокол предусматривает введение метки для контроля качества обслуживания и предотвращения фрагментации в промежуточных узлах. В этом протоколе предусматривается встроенное средство для аутентификации и шифрования данных.

Заголовок имеет длину 40 байт.

1. *Версия*. 4 бита.
2. *Приоритет*. Приоритет дейтаграммы. 4 бита.
3. *Метка потока*. Длина 24 бита. При помощи этого поля помечаются дейтаграммы, для которых в маршрутизаторах сети требуется специальная обработка.
4. *Длина всей IP-дейтаграммы минус длина заголовка*. 16 бит.
5. *Следующий заголовок*. Его длина 8 бит. В нём определяется заголовок, который находится за заголовком IP. Следующим заголовком может быть заголовок транспортного уровня либо заголовок расширения IP.
6. *Предельное число транзитов*. Длина 8 бит.
7. *Адрес источника*. 128 бит.
8. *Адрес получателя*. 128 бит.

3.1. Заголовки расширений IP

1. Заголовок параметров транзита. В нём содержится дополнительная информация для маршрутизаторов. Используется в настоящее время для передачи пакетов длиной до 4 ГБ.
2. Заголовок параметров адресата. В нём содержится информация, которую будет обрабатывать конечный получатель пакета.
3. Заголовок маршрутизации. Используется для маршрутизации. В нём содержится список узлов, через которые должна пройти IP-дейтаграмма. Он начинается: сначала указывается поле следующего заголовка, затем указывается длина заголовка маршрутизации, потом указывается тип маршрутизации, потом оставшиеся сегменты, т. е. оставшиеся узлы, через которые должна пройти дейтаграмма. После этого указывается сам маршрут.
4. Заголовок фрагментации. Используется при необходимости фрагментации дейтаграмм. Фрагментация может быть выполнена только отправителем. Заголовок состоит:

- следующий заголовок;
- смещение фрагмента. Длина 13 бит. Смещение измеряется в единицах, кратных 64 битам;
- 2 бита не используются;
- флаг “есть ещё фрагменты”;
- идентификатор дейтаграммы. Длина 32 бита.

5. Заголовок аутентификации.
6. Заголовок шифрования.

4. Протокол IPsec

Протокол IPsec обеспечивает защиту обмена данными в сетях за счёт шифрования и (или) аутентификации всего потока данных на уровне IP.

Протокол может работать в двух режимах:

1. Транспортный. В этом режиме защищаются только данные из IP-дейтаграмм, а заголовок IP-дейтаграммы не защищается.
2. Туннельный. В этом режиме защищается вся IP-дейтаграмма. Для этого к защищённой IP-дейтаграмме добавляется новый IP-заголовок, никак не защищённый. Обычно в нём указывается IP-адрес маршрутизатора или шлюза, который стоит в сети конечного получателя.

IPsec поддерживает два протокола защиты:

1. Аутентификация AH.
2. Протокол шифрования аутентификации ESP.

Внутри каждого из этих протоколов может использоваться несколько различных алгоритмов.

Дополнительно в протоколе IPsec определён протокол распределения ключей.

Заголовок аутентификации обеспечивает аутентификацию IP-дейтаграмм и проверку целостности данных в нём.

Заголовок состоит из следующих полей:

1. Следующий заголовок. Длина 8 бит.
2. Длина. Здесь длина заголовка в 32-битных единицах минус 2.
3. Зарезервированных 16 бит.
4. Индекс параметров защиты. Длина 32 бита. Он идентифицирует защищённую связь.
5. Порядковый номер. Длина 32 бита. Порядковый номер дейтаграммы, который был послан по данной защищённой связи.
6. Данные аутентификации. В нём содержится код аутентификации.

4.1. Защищённая связь

Связь — односторонние отношения между отправителем и получателем. Связь определяется параметрами:

1. Индекс параметров защиты. Строка битов, которая обозначает некий условный номер этой связи. По нему определяются алгоритмы обработки принятого пакета.
2. IP-адрес получателя.
3. Идентификатор протокола защиты. Параметры защищённой связи хранятся в специальных таблицах. В этих таблицах записаны:
   - счётчик порядкового номера;
   - флаг переполнения счётчика порядкового номера;
   - окно защиты от воспроизведения. Для защиты от повторной передачи одних и тех же дейтаграмм.
4. Информация AH. Хранятся параметры для алгоритма аутентификации.
5. Информация ESP. В нём хранятся параметры выбранного алгоритма шифрования.
6. Время жизни защищённой связи. Это интервал времени или значение счётчика байтов, по достижении которого связь уничтожается.
7. Режим IPsec.
8. Максимальная единица передачи маршрута. Максимальный размер пакета, который может быть передан без фрагментации.

Защищённые связи связываются с потоком IP через селекторы. Эти селекторы хранятся в базе данных политики защиты. Деление потоков может осуществляться по IP адресам пункта назначения, IP адресам источников, по протоколу транспортного уровня, по метке потока протокола IPv6 и т. п.

4.2. Формат пакетов ESP

1. Индекс параметров защиты. Длина 32 бита. Номер защищённой связи.
2. Порядковый номер дейтаграммы. Длина 32 бита.
3. Передаваемые данные.
4. Заполнитель. Нужен для правильной работы алгоритма шифрования.
5. Длина заполнителя.
6. Следующий заголовок. Длина 8 бит.
7. Данные аутентификации. Вычисляется для всей дейтаграммы ESP.

Шифры RC5, тройной DES, IDEA, BlowFish, CAST.

4.3. Управление ключами

Управление ключами может быть ручное (когда администратор сам вводит ключи в систему) и автоматизированное. Для автоматизированного применяя протокол ISAKMP/OAKLEY. OAKLEY — протокол управления ключами основан на алгоритме Диффи-Хеллмана.

К обычному Диффи-Хеллману в нём добавлена аутентификация сторон, обменивающихся ключами. Аутентификация может быть выполнена с помощью ЭЦП или алгоритмов шифрования.

4.4. Протокол ISAKMP

Протокол защищённой связи и управления ключами. Сообщения этого протокола состоят из заголовка и данных. Они передаются с помощью протокола транспортного уровня UDP. В заголовке присутствуют следующие поля:

1. Случайное число, которое генерируется стороной, изменяющей, создающей, удаляющей связь.
2. Случайное число объекта получателя.
3. Следующий полезный груз. В этом поле указывается тип данных, которые передаются в сообщении ISAKMP.
4. Главный номер версии.
5. Дополнительный номер версии.
6. Тип обмена.
7. Флаги. Флаг указывает зашифрованы или нет данные ISAKMP.
8. Бит фиксации. Он нужен, чтобы удостовериться, что сначала была создана защищённая связь, а потом получены соответствующие пакеты ISAKMP.
9. Универсальный идентификатор сообщения.
10. Длина сообщения в байтах.

Типы полезного груза:

1. Защищённая связь. Нужна, чтобы начать процесс создания защищённой связи.
2. Тип предложения. В нём указывается применяемый протокол ESP/AH, число трасформаций.
3. Трасформация. В каждой трасформации передаются атрибуты используемого алгоритма шифрования или аутентификации. Трансформаций может быть указано несколько.
4. Тип обмена ключами.
5. Идентификация. Предназначена для аутентификации связывающих сторон.

6. Сертификат. Сертификат открытого ключа.
7. Цифровая подпись.
8. Хеширование.
9. Запрос сертификата.
10. Нонс. Случайное число. Оно нужно, чтобы обеспечить защиту от атак воспроизведения сообщений и обеспечить процесс обмена сообщениями в реальном времени.
11. Тип уведомления.
12. Тип удаления. Удаление защищённой связи.

4.5. Тип обмена

1. Базовый обмен. Происходить обмен ключами и данными аутентификации одновременно.
2. Обмен с защитой идентификации сторон.
3. Обмен только данными аутентификации.
4. Обмен без идентификации сторон.
5. Информационный обмен. Нужен для передачи сообщений о параметрах управления защищённой связью.

5. Протоколы транспортного уровня

5.1. TCP

Протокол TCP является пакетным. Пакеты называются сегментами. Каждый сегмент имеет заголовок.

Формат *заголовка*:

1. Порт отправителя. Длина 16 бит.
2. Порт получателя. Длина 16 бит.
3. Позиция сегмента.
4. Первый ожидаемый байт. Используется только, если сегмент — это квитанция.
5. Смещение данных. Это длина заголовка в 32-разрядных словах. Длина 4 бита.
6. 6 бит неиспользуемых.
7. Флаги. 6 флагов.

    - URG. Срочность данных.
    - ACH. Квитанция.
    - PSH. Сегмент послать в первую очередь.

- RST. Запрос на установку первоначальных параметров соединения.
- SYN. Синхронизация счётчиков переданных данных при установлении соединения.
- FIN. Отправлен последний бит сообщения.

8. Размер окна. В нём указывается сколько байт готов принять получатель.
9. Контрольная сумма. Длина 16 бит. Контрольная сумма вычисляется на весь сегмент + IP адреса отправителя и получателя, идентификатор протокола и длину сегмента.
10. Указатель срочности данных.
11. Опции.

    - 0 — конец списка опций.
    - 1 — отсутствие операции.
    - 2 — максимальный размер сегмента.

12. Заполнитель. Дополняет заголовок до целого числа 32-разрядных слов.
13. Поле данных. Размер не фиксирован (максимальный указан в опции максимальный размер сегмента).

Номер порта — число, которое однозначно определяет приложение, осуществляющее сетевой обмен. Каждому приложения записан определённый номер порта.

Сокет — число, в которое входит IP адрес компьютера и номер порта. Однозначно определяет связь между процессами через протокол TCP.

Т. к. TCP отвечает за гарантированную доставку сообщений, поэтому передача сообщения происходит после установления соединения между отправителем и получателем. На каждую переданную дейтаграмму (сегмент) получатель должен послать квитанцию

5.1.1. Передача сообщения

Сообщение от прикладного уровня является потоком, представляет собой последовательность байт фиксированной длины, передаваемых асинхронно. TCP разбивает этот поток на сегменты и к каждому из них добавляет соответствующий заголовок. Длина сегмента задаётся администратором или определяется автоматически протоколом TCP.

Сначала происходит установление соединения. Отправитель посылает сегмент, в котором содержится номер сокета. В заголовке флаг SYN установлен в единицу. В ответ получатель посылает номер своего сокета. При этом в заголовке установлены флаги SYN и ASK. Отправитель посылает сегмент, в заголовке которого флаг ASK установлен в 1 и в поле, где указывается номер сегмента устанавливается 1. На этом процесс соединения заканчивается.

Если сообщение состоит из нескольких TCP сегментов, то получатель собирает его согласно порядковых номеров, хранящихся в заголовке. Если сегмент потерян или повреждён, то отправителю посылается сообщение, содержащее порядковый номер этого сегмента. Отправитель повторно передаёт запрошенный сегмент. Если сообщение принято, то посылается квитанция.

В последнем сегменте сообщения в заголовке должен быть установлен флаг FIN. После этого соединение разрывается.

Чтобы предотвратить переполнение буфера получателя используется т. н. скользящее окно, т. е. в заголовке передаётся размер окна, который может принять получатель.

В протоколе TCP используется несколько таймеров:

1. Таймер повтора передачи. Устанавливает время ожидания квитанции. Если квитанция за этот промежуток времени не поступает, сегмент считается потерянным и отправляется вновь. Повторная передача происходит заданное число раз. Если передача не удалась, то на прикладной уровень сообщается об ошибке.
2. Таймер задержки. Нужен, чтобы исключить повторное открытие только что закрытого порта, которое может быть вызвано прибывшими сегментами. Задержка может достигать 30 сек.
3. Таймер запроса. Нужен когда получатель приостановивший приём данных отправляет сообщение о возобновлении работы, но не получает подтверждения. Чтобы продолжить передачу, отправитель посылает запросы с периодом, заданным этим таймером.
4. Таймер контроля. Он вызывает периодическую передачу сегментов без данных. Нужен для проверки сети. Значение между 5–45 секундами.
5. Таймер разъединения. Задаёт максимальное время ожидания ответа. По истечении этого срока соединение разрывается. Максимальное время обычно равно 360 сек.

5.2. UDP

Это протокол транспортного уровня. Передача данных в нём происходит без установления соединения. Отправителю никак не сообщается доставлено ли его сообщение, правильно ли оно принято. Исправление ошибок происходит либо на сетевом, либо на прикладном уровне. Управление потоком данных не предусмотрено.

Заголовок UPD дейтаграммы:

1. Порт отправителя. Длина 16 бит. Поле необязательное.
2. Порт получателя. Длина 16 бит. Поле обязательно.
3. Длина дейтаграммы. Длина 16 бит.
4. Контрольная сумма. Длина 16 бит. Вычисляется также, как в протоколе TCP.
5. Данные.

6. Дополнительная лекция

6.1. IP адресация

IP адрес является уникальным 32-битным идентификатором IP интерфейса в сети Интернет, т. е. если у хоста несколько интерфейсов, у него будет несколько IP адресов.

IP адрес принято записывать в десятичном виде с разбивкой 32-битного числа по октетам. IP адрес состоит из 2 частей. Старшие разряды являются адресом сети, младшие разряды — адресом хоста. Граница разделов 2 частей определяются маской (subnet mask).

Маска — 32-битовая комбинация, в которой единицы установлены на сетевой части адреса, а нули на хостовой.

6.2. Классовая модель адресов

Существуют 5 классов адресов:

1. A. 255.0.0.0. Диапазон 0.0.0.0 - 127.0.0.0
2. B. 255.255.0.0. Диапазон 128.0.0.0 - 192.255.0.0
3. C. 255.255.255.0. Диапазон 193.0.0.0 - 223.255.255.255.0
4. D. Сеть мультиадресной рассылки. Адреса этого диапазона могут быть присвоены нескольким сетевым интерфейсам. Диапазон 224.0.0.0 - 239.0.0.0
5. E. Диапазон 240.0.0.0 - 255.255.255.255

6.3. Специальные IP адреса

- 0.0.0.0 — маршрут по умолчанию (default road). Используется в маршрутных таблицах для указания направления передачи пакетов, адресат которых неизвестен.
- 255.255.255.255 — широковещательный адрес (broadcast) локальной сети, в которой абонент находится.
- адрес, у которого хостовая часть нулевая называется адресом сети и он не может быть присвоен никакому хосту.
- адрес, у которого хостовая часть единицы называется широковещательным адресом удалённой сети. Он не может быть присвоен хосту.
- 127.0.0.0 — сеть обратной связи (loopback). В ней определён только один интерфейс — 127.0.0.1. Любой пакет, отправленный по адресу 127.0.0.1 будет принят этим же узлом так, как если бы он пришёл из сети. Используется для отладки сетевых сервисов без подключения к реальной сети.

6.4. Серые адреса

Любой пакет, отправленный по серому адресу будет отброшен маршрутизаторами сети Интернет и останется в пределах локальной сети. Поэтому адреса из серых диапазонов могут иметь несколько хостов в разных локальных сетях.

- A: 10.0.0.0
- B: 172.16.0.0 - 172.32.0.0
- C: 192.168.0.0 - 172.168.255.0

Для доступа с серого адреса к сети Интернет используется специальное устройство — прокси сервер, которое реализует функции трансляции адресов NAT.

6.5. Бесклассовая модель

1000000 128
1100000 192
1110000 224
1111000 240
1111100 248
1111110 254

Для получения адреса сети необходимо IP адрес узла в двоичном виде поразрядно умножить на маску. Для получения адреса хоста IP адрес в двоичном виде поразрядно необходимо умножить на инвертированную маску

Сеть 172.16.40.0/24
3 сети 20 хостов
172.16.40.000/00000
172.16.40.001/00000
172.16.40.010/00000
Диапазон 0: 172.16.40.000/00001 = 1
172.16.40.000/11110 = 30
1: 172.16.40.001/00001 = 33
172.16.40.001/11110 = 62
2: 172.16.40.010/00001 = 65
172.16.40.010/11110 = 94

6.6. Практика

65.179.19.241
255.255.128.0 маска
Найти адрес сети, хоста, диапазон, широковещательный адрес сети.
Умножить 19 на 128 в двоичном виде поразрядно.
Адрес сети: 65.179.0.0
128 инвертировать и умножить.

Адрес хоста: 0.0.19.241
Адрес сети: минимальный 65.179.00000000.0000001
максимальный 65.179.127.255
Широковещательный: 65.179.0.255

6.7. Маршрутизация

Адрес пол.	Маска пол.	Маршрутизатор	Интерфейс	Метрика
172.16.40.0	255.255.255.224	172.16.40.1	172.16.40.1	1
172.16.40.32	255.255.255.224	172.16.40.33	172.16.40.33	1
172.16.40.64	255.255.255.224	172.16.40.65	172.16.40.65	1
0.0.0.0	0.0.0.0	65.137.80.1	65.137.80.11	1

7. Протокол SSL

Протокол SSL предназначен для защищённой передачи данных через протокол TCP. Протокол SSL в стеке протоколов находится над TCP, но ниже протоколов прикладного уровня.

Протокол прикладного уровня, который используют SSL — HTTP.

SSL состоит из 4 отдельных протоколов: *Протокол записи*. Этот протокол непосредственно взаимодействует с протоколом TCP. Он обеспечивает конфиденциальность сообщений и целостность сообщений. Конфиденциальность обеспечивается за счёт шифрования данных, а целостность сообщений за счёт добавления к данным кода аутентичности (контрольная сумма). Данные приложения разбиваются на фрагменты размером не более 2^{14} байт. Потом может быть выполнено сжатие данных. На эти данные вычисляется код аутентичности и этот код добавляется после данных.

Сформированный блок шифруется с использованием симметричного алгоритма. К полученному зашифрованному блоку добавляется заголовок протокола записи. Сформированный пакет поступает на уровень протокола TCP.

Алгоритмы шифрования DES. Код аутентичности SHA-1.

Поля заголовка:

1. Тип содержимого. Длина 8 бит. Определяется протокол лежащего выше уровня, которому адресован фрагмент.
2. Главный номер версии. Длина 8 бит. Главный номер версии используемого протокола SSL.
3. Дополнительное поле. Длина 8 бит.
4. Дополнительный номер протокола.
5. Длина сжатого фрагмента. Длина 16 бит. Длина в байтах фрагмента открытого текста. Максимальное значение $2^{14} + 2048$.

Протокол изменения параметров шифрования. Этот протокол расположен над протоколом записи. Этот протокол служит для передачи сообщения

с параметром скопировать состояние ожидания в текущее состояние в результате чего обновляются параметры шифров, используемых для данного соединения. Сообщение представляет собой 00000001.

Протокол извещения. Он предназначен для обмена служебными сообщениями о работе SSL. Он также расположен над протоколом записи. Сообщение состоит из 2 байт. Первый байт означает уровень предупреждения или уровень неустранимой ошибки. Если уровень равен 2, то соединение по протоколу SSL разрывается. Второй байт — код, означающий смысл извещения.

Протокол квитирования. Лежит над протоколом записи. По этому протоколу происходит взаимная аутентификация сторон, согласовываются алгоритмы шифрования и их параметры, алгоритмы вычисления кода аутентичности, передаются ключи. Этот протокол используется до начала передачи данных. Сообщение протокола состоит из 3 частей:

1. Тип сообщения. Размер 1 байт.
2. Длина сообщения. Размер 3 байта.
3. Содержимое.

7.1. Работа протокола SSL

Определение характеристик защиты. Процесс передачи инициируется клиентом. Для этого он отправляет серверу сообщение “clien hello”. В качестве параметров этого сообщения указываются наивысший номер версии протокола, поддерживаемый клиентом; случайное число — используется во время обмена ключами для защиты от атак воспроизведения; идентификатор сеанса — сеанс в протоколе ssl это связь между клиентом и сервером; список шифров, которые поддерживает клиент; список методов сжатия, которые поддерживает клиент. В ответ на это сообщение сервер должен послать сообщение “server hello”, в котором будут те же параметры. В качестве случайного числа передаётся число, сгенерированное сервером.

Возможные методы обмена ключами — RSA, Диффи-Хелмана и его различные модификации.

Алгоритмы шифрование RC4, RC2, DES, IDEA, 3DES, DES40.

Вычисление кода проверки целостности MD5, SHA-1.

Аутентификация и обмен ключами. Если требуется аутентификация сервера, сервер отправляет свой сертификат X.509. Затем передаётся необходимая информация для выработки общего сеансового ключа с клиентом. В конце этой информации сервер отправляем “server done”. Получив это сообщение клиент проверяет подлинность сертификата. После клиент отправляет свой сертификат серверу и необходимую со своей стороны информацию для выработки общего сеансового ключа.

Завершение создания защищённого соединения. Клиент отправляет сообщение изменение параметров шифрования, т. е. начинает работать протокол изменения параметров шифрования.

Затем сразу же отправляется сообщение “finished”, которое зашифровано на выбранном алгоритмом с сформированным сеансовым ключом. В ответ на эти два сообщения сервер посылает своё сообщение изменение параметров шифрования и своё сообщение “finished”, зашифрованное при помощи ключа. Эти сообщения нужны для того, чтобы узнать, что у клиента и сервера один сеансовый ключ.

Если установление сеанса завершилось успешно, начинается передачи данных от протоколов вышележащих уровней.

7.2. Уязвимости SSL

1. Вскрытие используемых алгоритмов шифрования.
2. Уязвимость к атакам открытого текста. Эта атака используется для определения сеансового ключа. Она может успешна из-за того в открытом тексте часто встречаются одни и те же команды протокола HTTP.
3. Атака воспроизведения. Противник пытается передать серверу заранее перехваченное сообщение от клиента. Успешность данной атаки определяется длиной случайного числа, являющегося идентификатором сеанса.
4. Атака “посредник”. Противник между клиентом и сервером.

8. Протокол Kerberos

Этот протокол предназначен для аутентификации и обмена ключами, которые нужны для установки защищённого канала связи между абонентами, работающими в Интернете.

Этот протокол является протоколом прикладного уровня и разработан для сетей TCP/IP.

Kerberos состоит:

1. Сервер аутентификации.
2. Сервер выдачи мандатов.
3. Клиенты.
4. Серверы, к которым пользователи (клиенты) обращаются за каким либо ресурсом.

Подстроен на основе протокола Нитхема/Шредера с 3-ей доверенной стороной. Первый 2 компонента являются этой стороной.

На сервере аутентификации хранится БД, в которой записаны идентификаторы всех пользователей сети и их секретные ключи, идентификаторы всех ресурсов сети и их секретные ключи. Эти секретные ключи позволяют шифровать сообщения для клиентов и серверов. Успешное расшифрование этих сообщений является гарантией прохождения аутентификации всеми участниками протокола.

8.1. Описание протокола

1. Клиент посылает серверу аутентификации сообщения с запросом на разрешение доступа к серверу выдачи мандатов. Это сообщение включает в себя:

   - идентификатор клиента,
   - идентификатор сервера выдачи мандатов и
   - метку времени.

2. Сервер аутентификации отвечает клиенту в сообщении, которое зашифровано секретным ключом клиента, который хранился в БД. В этом сообщении содержится:

   - сеансовый ключ для связи с сервером выдачи мандатов,
   - идентификатор сервера выдачи мандатов,
   - метку времени, когда был отправлен ответ, срок действия мандата и мандат сервера выдачи мандата. *Мандат* — специальная информация, на основе которой происходит проверка подлинности обращающего субъекта.

3. Клиент посылает полученный мандат и идентификатор требуемого ему сервиса серверу выдачи мандатов. В этом сообщении присутствует *аутентификатор клиента*. Аутентификатор клиента из идентификатора клиента, сетевого адреса и метки времени. Эти 3 компонента зашифровываются на ключе, который был получен на шаге 2.
4. Сервер выдачи мандатов расшифровывает полученный аутентификатор клиента, проверяет разрешён ли клиенту доступ к запрашиваемому ресурсу и если разрешён, посылает сообщение, зашифрованное тем же ключом, полученным в шаге 2. Сообщение состоит из:

   - ключа для установления связи между клиентом и запрашиваемым сервисом,
   - идентификатора сервиса,
   - метки времени и мандата сервиса. *Мандат сервиса* — зашифрованное сообщение при помощи ключа связи между сервисом и сервером выдачи мандатов. Внутри этого сообщения находится:
     - сеансовый ключ для связи клиента и сервиса,
     - идентификатор клиента,
     - сетевой адрес клиента,
     - идентификатор сервиса,
     - метка времени,
     - срок действия мандата.

5. Клиент передаёт сервису полученный на шаге 4 мандат и свой аутентификатор. Аутентификатор на этом шаге шифруется при помощи ключа, полученного на шаге 4.

6. Сервис проверяет полученное сообщение. Если процедуры расшифрования прошли успешно, то аутентификация прошла успешно, т. е. сервис удостоверился, что к нему обращается клиент, указанный в сообщении. Происходит, если требуется аутентификация сервиса. Сервис отсылает клиенту зашифрованное сообщение, полученную на шаге 5 метку времени + 1-ца. Используется ключ тот же, что и на шаге 5 для шифрования аутентификатора.

Среда Kerberos должна удовлетворять следующим условиям:

1. Сервер аутентификации должен хранить свои байты данных, хешированные пароли всех пользователей системы. Пароли пользователей используются для формирования идентификатора клиента.
2. Все сервисы в сети должны быть зарегистрированы и у каждого из них должен быть свой секретный ключ для связи с сервером Kerberos.

Kerberos работает в пределах одной локальной сети. Если пользователю требуются ресурсы другой сети, то Kerberos доступа к ним не разрешит. Чтобы это устранить необходимо, чтобы оба сервера Kerberos были зарегистрированы друг в друге. Соответственно для связи между серверами должны быть установлены секретные ключи. При такой конфигурации клиент сначала в своей сети должен получить мандат доступа к серверу выдачи мандатов в другой сети. После этого клиент обращается к серверу выдачи мандатов другой сети и получает доступ к интересующему ресурсу.

Отличия 4 и 5 версии Kerberos: в версии 4 использовался алгоритм шифрования DES. В версии 5 может быть выбран любой другой алгоритм шифрования. В версии 4 требуется использование IP-адрасации, в версии 5 — любые сетевые адреса. В версии 4 срок действия мандата составлял 1280 минут максимум, потому, что срок действия мандата представлялся 8-битовым числом; в версии 5 явно указывается момент начала действия мандата и момент его окончания.

Уязвимости Kerberos:

1. Повторное использование перехваченной информации.
2. Синхронизация часов. Т. е. система будет работать правильно, если у всех её участников часы синхронизированы.
3. Сложность паролей.
4. Повторное использование идентификаторов субъектов. Новый объект системы может получить идентификатор выбывшего.
5. Сеансовые ключи. Один и тот же ключ используется в нескольких сеансах связи.

Kerberos в настоящее время поддерживается Windows и FreeBSD.