Лекция 10

Протоколы FTP, NTP.

FTP (File Transfer Protocol) — стандартный протокол, предназначенный для передачи файлов по TCP-сетям.

Протокол построен на архитектуре "клиент-сервер" и использует разные сетевые соединения для передачи команд и данных между клиентом и сервером. Пользователи FTP могут пройти аутентификацию, передавая логин и пароль открытым текстом, или же, если это разрешено на сервере, они могут подключиться анонимно. Можно использовать протокол SSH для безопасной передачи, скрывающей (шифрующей) логин и пароль, а также шифрующей содержимое.

FTP является одним из старейших прикладных протоколов, появившимся задолго до HTTP, и даже до TCP/IP, в 1971 году. В первое время он работал поверх протокола NCP[1].

Особенность протокола FTP в том, что он использует множественное (как минимум - двойное) подключение. При этом один канал является управляющим, через который поступают команды серверу и возвращаются его ответы (обычно через TCP-порт 21), а через остальные происходит собственно передача данных, по одному каналу на каждую передачу. Поэтому в рамках одной сессии по протоколу FTP можно передавать одновременно несколько файлов, причём в обоих направлениях. Для каждого канала данных открывается свой TCP порт, номер которого выбирается либо сервером, либо клиентом, в зависимости от режима передачи.

Протокол FTP имеет двоичный режим передачи, что сокращает накладные расходы трафика и уменьшает время обмена данными при передаче больших файлов. Протокол же HTTP обязательно требует кодирования двоичной информации в текстовую форму, например при помощи алгоритма Base64.

Протокол определен в **RFC 959**. Сервер отвечает по потоку управления трехзначными ASCII-кодами состояния с необязательным текстовым сообщением. Например, "200" (или "200 OK") означает, что последняя команда была успешно выполнена. Цифры представляют код ответа, а текст - разъяснение или запрос. Текущая передача по потоку данных может быть прервана с помощью прерывающего сообщения, посылаемого по потоку управления.

FTP может работать в *активном или пассивном режиме*, от выбора которого зависит способ установки соединения. В активном режиме клиент создаёт управляющее TCP-соединение с сервером и отправляет серверу свой IP-адрес и произвольный номер клиентского порта, после чего ждёт, пока сервер не запустит TCP-соединение с этим адресом и номером порта. В случае, если клиент находится за брандмауэром и не может принять входящее TCP-соединение, может быть использован пассивный режим. В этом режиме клиент использует поток управления, чтобы послать серверу команду PASV, и затем получает от сервера его IP-адрес и номер порта, которые затем используются клиентом для открытия потока данных с произвольного клиентского порта к полученному адресу и порту.

---

1 **The Network Control Protocol (NCP)** — сетевой протокол, который был первым стандартом сетевого протокола в ARPANET. NCP был завершен и представлен в декабре 1970 г. компанией Network Working Group (NWG), возглавляемой Стивом Крокером — изобретателем Request For Comments.

При передаче данных по сети могут быть использованы четыре представления данных:

1. *ASCII* - используется для текста. Данные, если необходимо, до передачи конвертируются из символьного представления на хосте-отправителе в "восьмибитный ASCII", и (опять же, если необходимо) в символьное представление принимающего хоста. Как следствие, этот режим не подходит для файлов, содержащих не только обычный текст.
2. *Режим изображения (обычно именуемый бинарным)* - устройство-отправитель посылает каждый файл байт за байтом, а получатель сохраняет поток байтов при получении. Поддержка данного режима была рекомендована для всех реализаций FTP.
3. *EBCDIC* - используется для передачи обычного текста между хостами в кодировке EBCDIC. В остальном, этот режим аналогичен ASCII-режиму.
4. *Локальный режим* - позволяет двум компьютерам с идентичными установками посылать данные в собственном формате без конвертации в ASCII.

Для текстовых файлов предоставлены различные форматы управления и настройки структуры записи. Эти особенности были разработаны для работы с файлами, содержащими Telnet или ASA-форматирование.

Передача данных может осуществляться в любом из трёх режимов:

1. *Поточный режим* - данные посылаются в виде непрерывного потока, освобождая FTP от выполнения какой бы то ни было обработки. Вместо этого, вся обработка выполняется TCP. Индикатор конца файла не нужен, за исключением разделения данных на записи.
2. *Блочный режим* - FTP разбивает данные на несколько блоков (блок заголовка, количество байт, поле данных) и затем передаёт их TCP.
3. *Режим сжатия* - данные сжимаются единым алгоритмом (обыкновенно, кодированием длин серий).

Аутентификация

FTP-аутентификация использует обычную схему имя пользователя/пароль для предоставления доступа. Имя пользователя посылается серверу командой USER, а пароль - командой PASS. Если предоставленная клиентом информация принята сервером, то сервер отправит клиенту приглашение и начинается сессия. Пользователи могут, если сервер поддерживает эту особенность, войти в систему без предоставления учётных данных, но сервер может предоставить только ограниченный доступ для таких сессий.

Анонимный FTP

Хост, обеспечивающий FTP-сервис, может предоставить анонимный доступ к FTP. Пользователи обычно входят в систему как "anonymous" (может быть регистрозависимым на некоторых FTP-серверах) в качестве имени пользователя. Хотя обычно пользователей просят прислать адрес их электронной почты вместо пароля, никакой проверки фактически не производится. Многие FTP-хосты, предоставляющие обновления программного обеспечения, поддерживают анонимный доступ.

FTP не разрабатывался как защищённый (особенно по нынешним меркам) протокол и имеет многочисленные уязвимости в защите.

FTP не может зашифровать свой трафик, все передачи - открытый текст, поэтому имена пользователей, пароли, команды и данные могут быть прочитаны кем угодно, способным перехватить пакет по сети. Эта проблема характерна для многих спецификаций Интернет-протокола (в их числе SMTP, Telnet, POP, IMAP), разработанных до создания таких механизмов шифрования, как TLS и SSL. Обычное решение этой проблемы - использовать "безопасные", TLS-защищенные версии уязвимых протоколов (FTPS для FTP, TelnetS для Telnet и т.д.) или же другой, более защищённый протокол, вроде SFTP/SCP, предоставляемого с большинством реализаций протокола Secure Shell.

Существует несколько методов безопасной передачи файлов, в одно или другое время называемых **"Безопасным FTP"**.

FTPS

Явный FTPS - расширение стандарта FTP, позволяющее клиентам требовать того, чтобы FTP-сессия была зашифрована. Это реализуется отправкой команды "AUTH TLS". Сервер обладает возможностью позволить или отклонить соединения, которые не запрашивают TLS. Это расширение протокола определено в спецификации RFC 4217. Неявный FTPS - устаревший стандарт для FTP, требующий использования SSL- или TLS-соединения. Этот стандарт должен был использовать отличные от обычного FTP порты.

SFTP

SFTP, или «SSH File Transfer Protocol», не связан с FTP, за исключением того, что он тоже передаёт файлы и имеет аналогичный набор команд для пользователей. SFTP, или безопасный FTP, — это программа, использующая SSH (Secure Shell) для передачи файлов. В отличие от стандартного FTP он шифрует и команды, и данные, предохраняя пароли и конфиденциальную информацию от открытой передачи через сеть. По функциональности SFTP похож на FTP, но так как он использует другой протокол, клиенты стандартного FTP не могут связаться с SFTP-сервером и наоборот.
FTP через SSH (не SFTP)

FTP через SSH относится к практике туннелирования обычной FTP-сессии через SSH-соединение. Поскольку FTP использует несколько TCP-соединений, туннелирование через SSH особенно затруднительно. Когда много SSH-клиентов пытаются установить туннель для канала управления (изначальное "клиент-сервер" соединение по порту 21), защищён будет только этот канал; при передаче данных программное обеспечение FTP на любом конце установит новые TCP-соединения (каналы данных), которые обойдут SSH-соединение и, таким образом, лишатся целостной защиты.

Коды ответов FTP

Код ответа - трёхзначное число. Первая цифра отвечает за один из трёх исходов: успех, отказ или указание на ошибку либо неполный ответ.

2xx – Успешный ответ
4xx/5xx – Команда не может быть выполнена
1xx/3xx – Ошибка или неполный ответ

Вторая цифра определяет тип ошибки:

x0z – Синтаксическая.
x1z – Информация. Соответствует информационному сообщению.
x2z – Соединения. Сообщение относится к управляющему соединению либо к соединению данных.
x3z – Соответствует сообщениям об аутентификации пользователя и его правах.
x4z – Не определено.
x5z – Файловая система. Соответствует сообщению о состоянии файловой системы.

Третья цифра окончательно специфицирует ошибку.

FXP (File eXchange Protocol — протокол обмена файлами) — способ передачи файлов между двумя FTP-серверами напрямую, не закачивая их на свой компьютер. При FXP-сессии клиент открывает два FTP-соединения к двум разным серверам, запрашивая файл на первом сервере, указывая в команде PORT IP-адрес второго сервера.

Несомненным преимуществом поддержки стандарта FXP является то, что на конечных пользователей, желающих скопировать файлы с одного FTP-сервера на другой, уже не действует ограничение пропускной способности их собственного интернет-соединения. Нет необходимости скачивать себе файл, чтобы потом загрузить его на другой FTP-сервер. Таким образом, время передачи файлов будет зависеть только от скорости соединения между двумя удаленными FTP-серверами, которая в большинстве случаев заведомо больше «пользовательской».

FXP стал использоваться злоумышленниками для атак на другие серверы: в команде PORT указывается IP-адрес и порт атакуемого сервиса на компьютере жертвы, и командами RETR/STOR производится обращение на этот порт от лица FTP-сервера, а не атакующей машины, что позволяло устраивать масштабные DDoS-атаки с использованием сразу многих FTP-серверов, либо обходить систему безопасности компьютера жертвы, если он полагается только на проверку IP клиента и используемый для атаки FTP-сервер находится в доверенной сети или на шлюзе. В результате сейчас практически все серверы проверяют соответствие IP-адреса, указанного в команде PORT, IP-адресу FTP-клиента и по умолчанию запрещают использование там IP-адресов третьих сторон. Таким образом, использование FXP невозможно при работе с публичными FTP-серверами.

Network Time Protocol (NTP)

Network Time Protocol (NTP) — сетевой протокол для синхронизации внутренних часов компьютера с использованием сетей с переменной латентностью[2].

NTP использует для своей работы протокол UDP. Система NTP чрезвычайно устойчива к изменениям латентности среды передачи.

NTP использует алгоритм Марзулло (Keith Marzullo из Университета Калифорнии, Сан-Диего), включая такую особенность, как учёт времени передачи. В версии 4 способен достигать точности 10 мс (1/100 с) при работе через Интернет, и до 0,2 мс (1/5000 с) и лучше внутри локальных сетей.

NTP — один из старейших используемых протоколов. NTP разработан Дэвидом Л. Миллсом (David L. Mills) из университета Дэлавера в 1985 году и в настоящее время продолжает совершенствоваться. Текущая версия — NTP 4.

NTP использует иерархическую систему «часовых уровней» (stratum). Уровень 1 синхронизирован с высокоточными часами, например, с системой GPS, ГЛОНАСС (Единая Государственная шкала времени РФ) или атомным эталоном времени. Уровень 2 синхронизируется с одной из машин уровня 1, и так далее.

Время представляется в системе NTP 64-битным числом (8 байт), состоящим из 32-битного счётчика секунд и 32-битного счётчика долей секунды, позволяя передавать время в диапазоне 2^{32} секунд, с теоретической точностью 2^{-32} секунды. Поскольку шкала времени в NTP повторяется каждые 2^{32} секунды (136 лет), получатель должен хотя бы примерно знать текущее время (с точностью 50 лет). Также следует учитывать, что время отсчитывается с полуночи 1 января 1900 года, а не с 1970, поэтому из времени NTP нужно вычитать почти 70 лет (с учётом високосных лет), чтобы корректно совместить время с Windows или Unix-системами.

Более простая реализация этого алгоритма известна как SNTP — простой синхронизирующий сетевой протокол. Используется во встраиваемых системах и устройствах, не требующих высокой точности, а также в пользовательских программах точного времени.

NTP использует иерархическую, многоуровневую систему источников времени. Каждый уровень этой иерархии называется слоем, каждому слою присваивается номер, начиная с 0 (ноль) в верхней части. Уровень слоя определяет расстояние от эталонных часов и существует, чтобы предотвратить циклические зависимости в иерархии. Важно отметить, что слой не является показателем качества и надежности, это значит, что источник слоя 3 может дать сигнал более высокого качества, чем некоторые источники слоя 2. В основном, слои служат для распределения нагрузки и обеспечения большей площади покрытия. Это определение слоя также отличается от понятия часовых слоёв, используемых в телекоммуникационных системах.

1. **Слой 0.** Высокоточные приборы служащие эталоном времени, такие как атомные (молекулярные, квантовые) часы, радиочасы или их аналоги. Обычно эти устройства не подключены к сети; вместо этого они подключены к локальному компьютеру (например, через интерфейс RS-232) и передают сигналы PPS для синхронизации.
2. **Слой 1.** Это компьютер, к которому напрямую подключены эталонные часы. Он

2 Задержка или ожидание, которая увеличивает реальное время отклика по сравнению с ожидаемым.

выступает в качестве сетевого сервера времени и отвечает на NTP-запросы посылаемые компьютерами слоя 2.

3. **Слой 2.** Это компьютеры, которые получают время от серверов первого слоя, используя для этого протокол NTP. Обычно, компьютеры второго слоя обращаются к нескольким серверам первого слоя, и используя NTP-алгоритм, получают наилучший образец данных, отсеивая сервера с очевидно неверным временем. Компьютеры могут сравнивать свои данные с другими компьютерами своего слоя для получения стабильных и непротиворечивых данных на всех компьютерах слоя. Компьютеры второго слоя в свою очередь выступают в качестве серверов для компьютеров третьего слоя и отвечают на NTP-запросы.
4. **Слой 3.** Компьютеры третьего слоя работают точно так же как и компьютеры второго слоя, с той лишь разницей, что серверами для них являются компьютеры вышележащего второго слоя. Они так же могут выступать в качестве серверов для нижележащего слоя. NTP (в зависимости от версии) поддерживает до 256 слоев.