Протоколы, сервисы и услуги
в Интернет и IP-сетях

Тема № 11
Протоколы передачи файлов. FTP, TFTP. SFTP

доц. каф. СС и ПД, к.т.н. С. С. Владимиров

2017 г.

Протокол FTP

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 (Network Control Protocol — сетевой протокол, который был первым стандартом сетевого протокола в ARPAnet).

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.

Режимы работы FTP

Режимы работы FTP

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.

Режимы передачи данных в FTP

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).

Представление данных и аутентификация в FTP

Форматы представления данных в FTP

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 (бинарный режим) — 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.

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

FTP-сервер, может предоставить анонимный доступ к FTP. Традиционно для этого используются логины «anonymous» и «ftp». 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.

Безопасность в FTP

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.

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.

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.

Протокол FXP

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.

Протокол TFTP

TFTP (Trivial File Transfer Protocol — простой протокол передачи файлов)

Используется главным образом для первоначальной загрузки бездисковых рабочих станций. В отличие от FTP, не содержит возможностей аутентификации (хотя возможна фильтрация по IP-адресу) и основан на транспортном протоколе UDP.
TFTP был разработан в 1980 году и определен в RFC 1350 (STD 33). За ним закреплен 69 порт UDP.
Основное назначение TFTP — обеспечение простоты реализации клиента. В связи с этим он используется для загрузки бездисковых рабочих станций, загрузки обновлений и конфигураций в «умные» сетевые устройства, записи статистики с мини-АТС и аппаратных маршрутизаторов/файрволов.

Безопасность в TFTP

Поскольку протокол не поддерживает аутентификации, единственный метод идентификации клиента — это его сетевой адрес. Обычно в Unix-системах серверу `tftpd` доступен только каталог `/tftpboot`. Однако в старых TFTP-серверах было возможным получить файл паролей командой `RRQ ../etc/passwd`.
Демон `tftpd` (одна из реализаций tftp-сервера) отказывается обрабатывать файлы, содержащие в своём имени комбинацию «`/../`» или начинающуюся с «`../`». Запись разрешается только в файлы, которые уже существуют (любого размера, например нулевого), и доступны для публичной записи (права доступа: `-rw-rw-rw-`).
Дополнительная защита от доступа к произвольным файлам осуществляется с помощью смены корневого каталога на каталог `tftpd` (обычно `/usr/TFTPRoot`).

Протокол SCP

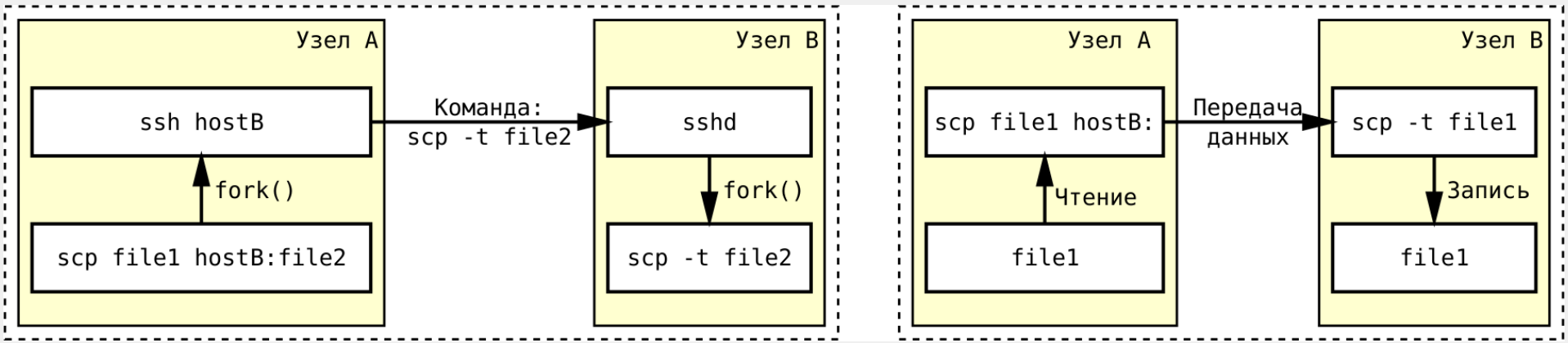
SCP (Secure Copy Protocol)

Сетевой протокол прикладного уровня (и обноименная утилита), предназначенные для передачи данных между двумя сетевыми узлами (локальным и удаленным или парой удаленных). Работает поверх протокола SSH, используя его механизмы аутентификации и защиты данных при передаче. Соответственно, так же использует 22 порт TCP.
SCP основан на протоколе/утилите RCP (Secure Copy), являвшейся частью пакета программ Berkeley r-commands (1982), который долгое время использовался в ОС Unix и стал стандартом де-факто для удаленного управления в Unix. R-commands не обеспечивали необходимый уровень безопасности (в частности, данные не шифровались при передаче) и были заменены на SSH вскоре после его появления.

Порядок работы SCP

SCP при передаче использует клиент-серверный принцип. Вначале на удаленной стороне запускается SCP в режиме записи или чтения файла (используются скрытые опции утилиты: `-t` и `-f`, соответственно), а затем производится подключение со стороны локальной машины. При работе с директорией целиком используется дополнительная скрытая опция `-d`.

Принцип передачи файла на удаленный хост



Протокол SCP. Режимы работы и команды

Режимы работы SCP на удаленном узле

1. Режим чтения (source mode). Соответствует флагу `-f`.
2. Режим записи (sink mode). Соответствует флагу `-t`.

Команды протокола SCP

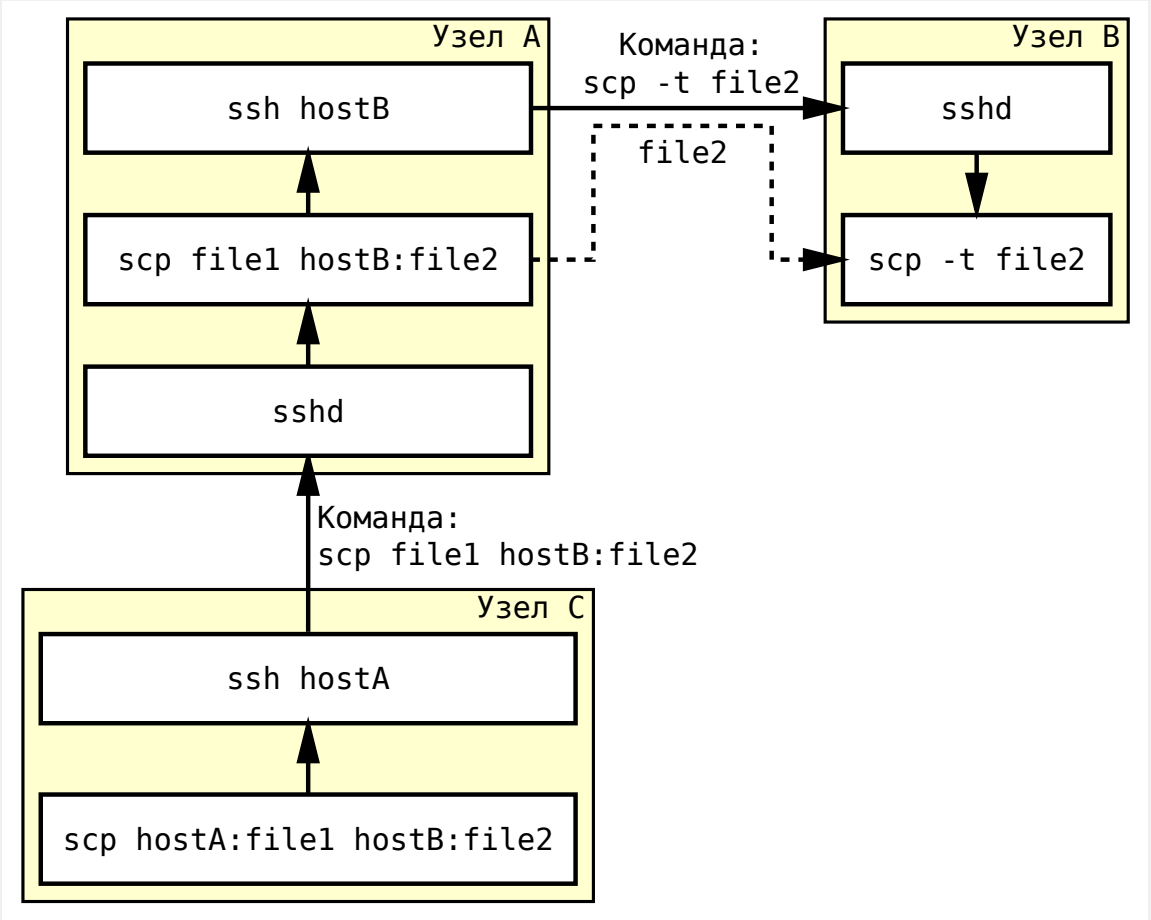
- `Cmmmm <length> <filename>` — Копирование одиночного файла: `mmmm` — права на файл; `<length>` — размер файла в байтах. Пример: `C0644 299 file1`. После этой команды должны следовать данные. Принимающая сторона считывает ровно столько данных, сколько указано в параметре `<length>`.
- `Dmmmm <length> <dirname>` — Рекурсивное копирование каталога. Параметр <length> игнорируется, но должен быть указан (обычно 0). Пример: `D0755 0 docs`. После этой команды должны следовать или команды передачи файла C, или команда E. На стороне «сервера» в этом случае необходимо использовать флаг «-r».
- `E` — Окончание рекурсивного копирования каталога.
- `T<mtime> 0 <atime> 0` — время доступа и изменения файла/директории в формате UNIX (в секундах от 00:00:00 UTC, 1 января 1970). Нули были введены на тот случай, если понадобится точность до микросекунд. Пример: `T1183828267 0 1183828267 0`. Передается до команд передачи файлов.

Размер передаваемых файлов

Поскольку в реализации протокола OpenSSH для параметра `<length>` используется переменная типа `long long int`, максимально можно передать 2^{63} байт (чуть больше $9 \cdot 10^{18}$ байт). Учитывая, что даже 2^{40} байт — это примерно 1 ТБ, можно считать, что ограничений на размер передаваемого файла нет.
При передаче большого количества небольших файлов (или передаче директории) оптимально использовать предварительную архивацию. Это значительно увеличивает скорость передачи по сети (даже без сжатия архива) за счет уменьшения накладных расходов на обработку каждого файла в отдельности.

Протокол SCP. Передача между удаленными узлами

Передача между удаленными узлами



При таком способе передачи не используется парольная аутентификация между узлами A и B. Допустимо использование host-based аутентификации или аутентификации по публичному ключу (незапароленному).

Протокол SFTP

SFTP (SSH File Transfer Protocol)

Протокол прикладного уровня, предназначенный для копирования и выполнения других операций с файлами поверх надёжного и безопасного соединения. Протокол разработан группой IETF как расширение к SSH-2, однако SFTP допускает реализацию и с использованием иных протоколов сеансового уровня.
Протокол предполагает, что он работает поверх установленного безопасного канала, что сервер уже аутентифицировал клиента и что идентификатор клиента доступен протоколу. Сервер SFTP обычно использует TCP порт 22.
SSH File Transfer Protocol не является протоколом FTP работающим поверх SSH — это другой, новый протокол. Также SFTP иногда путают с Simple File Transfer Protocol из-за совпадающего сокращения «SFTP».

В сравнении с другим протоколом, тоже предназначенным для копирования файлов поверх SSH — протоколом SCP, который позволяет только копировать файлы, SFTP даёт возможность выполнять намного больше операций с ними: например, докачивать файл после разрыва соединения или удалять файл на сервере и многие другие операции. По этой причине существуют графические и псевдографические клиенты для SFTP, но нет таких, кто использовал бы только SCP в чистом виде.

Сам по себе протокол SFTP не обеспечивает безопасность работы; это делает нижележащий протокол. Как правило, SFTP используется в сочетании с протоколом SSH2. Можно использовать SFTP и с другими протоколами, например SSH1, но это сопряжено с дополнительными трудностями.
Разработкой протокола занималась одна из групп IETF под названием Secsh — группа, ранее подготовившая стандарт SSH-2. Рабочая документация к новому протоколу SFTP не стала официальным стандартом, однако начала активно применяться для разработки приложений. В ходе развития протокола было выпущено шесть версий протокола. Постепенное наращивание функциональности протокола привело к тому, что 14 августа 2006 года было принято решение о прекращении работы над развитием протокола в связи с выполнением основной задачи проекта (разработка SSH) и отсутствием достаточного экспертного уровня для перехода к разработке полноценного протокола удалённой файловой системы.
Последней разработанной версией протокола является Draft 13 от 10 июля 2006 года.

Источники

- Материалы с сайта `https://wikipedia.org/`
- Telecommunication technologies — телекоммуникационные технологии / Ю. А. Семенов. URL: `http://book.itep.ru/`
- SFTP // Xgu.ru. URL: `http://xgu.ru/wiki/Sftp`
- How the SCP protocol works / Jan Pechanec. URL: `https://web.archive.org/web/20170215184048/https://blogs.oracle.com/janp/entry/how_the_scp_protocol_works`