Сетевые протоколы для удалённого управления компьютером

(WinFrame ,Windows Terminal Server, Telnet, SSH, rlogin и т.п.)

Режим дистанционного управления позволяет клиентам управлять работой удаленного сервера или рабочей станции по сети. Этот режим обычно используется сисадминами для администрирования выделенных серверов и технической поддержки пользователей.

Удаленный компьютер (удаленный хост) — сервер, физически размещенный на некотором расстоянии от клиента и доступ к которому (для клиентов) возможен только по сети.

Для подключения к удаленному хосту используются специализированные протоколы удаленного управления. Эти протоколы, реализующие модель «терминал-сервер», позволяют работать с ресурсами сервера так же, как и с локальными ресурсами: выполнять команды, работать с файловой системой, запускать приложения и т.п. С распространением персональных ЭВМ терминальный доступ стал использоваться в основном для решения задач удаленного администрирования.

Терминалы

С середины 1970-х до середины 1980-х годов вся работа за компьютером выполнялась через специальные устройства ввода-вывода — терминалы. С помощью терминалов пользователи подключались к серверу (мэйнфрейму) и использовали его системные ресурсы для решения прикладных задач.

Изначально, терминал — это оконечное сетевое устройство, подключенное к вычислительной системе и предназначенное для ввода и вывода данных. Команды, принимаемые с устройства ввода терминала (клавиатуры), передаются на удаленный сервер, где и выполняются. Результаты обработки возвращаются и отображаются на устройстве вывода терминала (дисплее). Впоследствии были разработаны эмуляторы терминалов — специальные программы, выполняющие те же задачи.

Можно выделить два основных типа терминалов:

1. Реальный, или физический терминал — как правило подразумевает устройство, вычислительные способности которого ограничены возможностью отображать то, что ему передано из сети (как максимум — полноэкранную графику).
2. Виртуальный терминал — сетевое приложение (программа), выполняющее функции физического терминала. Со стороны удаленного хоста такая программа, ничем не отличается от реального терминала. Однако терминалы различаются не только по представлению, но и по возможностям.

В зависимости от них различают:

- текстовые терминалы;
- графические терминалы;
- «интеллектуальные» терминалы.

Текстовые терминалы

Текстовый терминал, также известный как “последовательный терминал”, “ASCII-терминал”, “символьный терминал” и “видеодисплейный терминал” (VDT, video display terminal) — это устройство (или программа) ввода/вывода, работающее в дуплексном режиме и использующее символьно-ориентированные (байт-ориентированные) протоколы передачи данных, такие как telnet.

Потоки данных между терминалом и сервером представляют ASCII-символы. Специальные символы (или последовательности символов) представляют команды управления сервером и ответы сервера. Существуют и служебные команды, управляющие выводом (перевод строки, полужирный/наклонный текст, звуковой сигнал и т.п.) и сеансом связи (согласование версий протокола и др.).

Текстовые терминалы отличаются минимальными требованиями к полосе пропускания сети. Это позволяет дистанционно управлять удаленными серверами (например, в сети Интернет), используя медленные коммутируемые соединения (модемную связь через порт RS-232C).

«Продвинутые» модели текстовых терминалов поддерживают т.н. «блочный режим», основанный на буферизации вводимых данных. Символы, которые были получены с клавиатуры, сохраняются в памяти терминала (и их можно отредактировать встроенным строковым редактором) до нажатия клавиши ввода/передачи (Enter или что-то подобное). Таким образом, серверу передается целая строка (блок символов).

«Тупые» (dumb) терминалы

Есть множество различных определений “dumb-терминалов”, которые, суть, характеризуют максимальные возможностей терминала. Чаще всего под dumb-терминалами подразумевают текстовые терминалы, которые способны отображать на экране только алфавитно-цифровые символы.

Графические возможности текстовых терминалов ограничены специальными символами ASCII, которые могут быть использованы для рисования рамок таблиц, стрелок, горизонтальных и вертикальных линий, заливок разной плотности и т.д. Использование VGA палитры позволяет задавать цвет фона и текста. Но все это сводится к псевдографике, т.к. текстовые терминалы не управляют выводом на уровне отдельных пикселей

Графические терминалы

Графические терминалы, в отличие от текстовых, представляют возможности работы с полноцветной растровой и векторной графикой при вводе и отображении данных. Такие терминалы применяются в системах, поддерживающих графический пользовательский интерфейс (GUI, Graphic User Interface). Это позволяет получать данные от пользователя не только с клавиатуры, но и от точечных устройств ввода (“мышь”, тачпад, сенсорный экран и т.п.)

Для графических терминалов применяются два типа дисплеев:

- Растровые – где изображение строится построчно электронным лучем (ЭЛТ-дисплеи) или попиксельно (ЖК-дисплеи). Растровая графика используется в подавляющем большинстве случаев, в силу универсальности и приемлемой стоимости подобных устройств. Примерами протоколов графических терминалов являются RDP, ICA, VNC.
- Векторные – используют интеллектуальную электронику, чтобы перемещать электронный луч не только построчно, но и в любом направлении. Это позволяет строить высококачественные изображения без искажений на устройствах с разной разрешающей способностью. Основной недостаток векторных терминалов — их дороговизна, из-за чего они мало распространены.

Требования к пропускной способности линии связи при использовании графических терминалов, в общем случае, существенно выше по сравнению с текстовыми, т.к передавать приходится не коды символов, а, фактически, скриншоты. Для снижения нагрузки на сеть используются разные способы: компрессия, ограничения на глубину цвета и разрешение экрана, передача только изменившихся областей и др.

Текстовый режим работы также поддерживается графическими терминалами, но реализован иначе — внутри основного протокола.

«Интеллектуальные» терминалы

Под термином «интеллектуальный терминал» обычно подразумевают аппаратно-программный комплекс, ресурсов которого достаточно не только для вывода полноэкранной графики, но и для выполнения некоторых вычислений. Принципиальным отличием интеллектуального терминала от обычного ПК является то, что исполнимые программы должны загружаться с сервера, а выполняться на клиенте (самом терминале).

В приведенной трактовке типичным примером интеллектуального терминала является бездисковая рабочая станция — компьютер, у которого отсутствует жесткий диск (а зачастую и сменные накопители). Загрузка операционной системы и приложений для такого компьютера выполняется по сети.

«Тонкие клиенты»

«Тонкий клиент» (thin client) — разновидность интеллектуального терминала. Правильнее – его программное обеспечение, которое позволяет загружать и выполнять сетевые приложения (клиентские скрипты, Java-апплеты, ActiveX-приложения и т.п.) в некотором программном окружении (например, в браузере или виртуальной машине).

Протоколы удаленного управления

Telnet как протокол описан в RFC-854 (май, 1983 год). Его авторы J.Postel и J.Reynolds во введении к документу определили назначение telnet так: "Назначение TELNET-протокола -- дать общее описание, насколько это только возможно, двунаправленного, восьмибитового взаимодействия, главной целью которого является обеспечение стандартного метода взаимодействия терминального устройства и терминал-ориентированного процесса. При этом этот протокол может быть использован и для организации взаимодействий "терминал-терминал" (связь) и "процесс-процесс" (распределенные вычисления)."

Под telnet понимают триаду, состоящую из: telnet-интерфейса пользователя; telnetd-процесса; TELNET-протокола.

Эта триада обеспечивает описание и реализацию сетевого терминала для доступа к ресурсам удаленного компьютера.

Основными реализациями протокола Telnet являются:

Клиенты: telnet (Unix), PuTTY, telnet.exe (Windows),z/Scope Express VT,AbsoluteTelnet

Серверы: telnetd, MS Telnet.

В настоящее время из-за своей небезопасности протокол Telnet считается устаревшим и вместо него используется протокол SSH.

telnet

Служба удаленного управления telnet (Teletype Network)) — одна из самых старых сетевых технологий Internet (первая спецификация — RFC 158 от 19.05.1971 г.). Текущая спецификация telnet — RFC 854/STD 8 – Telnet Protocol Specification. По умолчанию сервер telnet принимает входящие подключения на 23-ий порт TCP (см. Сетевые сервисы).

Протокол telnet изначально разрабатывался для использования в гетерогенных сетях. В его основе — концепция сетевого виртуального терминала (Network Virtual Terminal, NVT) — механизма абстрагирования от специфики ввода/вывода различных аппаратных и программных платформ. Так, например, в UNIX в качестве символа перехода на другую строку используется LF (код 13), в то время как в MS-DOS и Windows — пара символов CR-LF (коды 10 и 13). Сетевой виртуальный терминал NVT предлагает использование унифицированного набора символов, который используется для преобразования кодов клиента и сервера.

Хотя в сессии telnet выделяют клиентскую и серверную сторону, протокол является симметричным и обе стороны взаимодействуют через NVT, обмениваясь данными двух типов:

- Прикладными данными (т.е. данными, которые идут от пользователя к серверному приложению и обратно);
- Опциями протокола telnet, служащими для уяснения возможностей и предпочтений сторон.

Прикладные данные проходят через протокол без изменений т.е. на выходе второго виртуального терминала мы видим именно то, что было введено на вход первого. С точки зрения протокола эти данные – просто последовательность байтов (октетов), по умолчанию принадлежащих набору ASCII. Если же установлена опция Binary — то последовательность любых данных в двоичном представлении.

Все значения октетов прикладных данных кроме \377 (десятичное 255) передаются по транспорту как есть. Октет \377 передаётся последовательностью \377\377 из двух октетов. Это связано с тем, что октет \377 используется для кодирования опций протокола.

Telnet поддерживает четыре режима передачи:

- Полудуплексный режим. NVT по умолчанию это полудуплексное устройство, которое требует исполнения специальной команды (GO AHEAD, GA) от сервера, перед тем как будет принят ввод от пользователя. Ввод пользователя отображается локальным эхом от NVT клавиатуры на NVT принтер, таким образом, от клиента к серверу посылаются только полные строки.
- Посимвольный. Каждый вводимый символ отправляется серверу отдельно от других. Сервер отражает эхом (опция ECHO — RFC 857) большинство символов и эти символы отражаются на экране клиента.
- Блочный. Отправка данных производится по принципу “строка за один раз”.
- Линейный режим (linemode). В данном случае этот термин означает реальную опцию linemode (RFC 1184). Эта опция обсуждается клиентом и сервером и корректирует все недостатки в режиме строка за один раз.

С точки зрения безопасности протокол telnet уязвим для любого вида атак, к которым уязвим его транспорт, т.е. протокол TCP. В протоколе не предусмотрено использование ни шифрования, ни проверки подлинности данных. Поэтому telnet можно использовать в полностью контролируемой сети или с применением защиты на сетевом уровне (различные реализации виртуальных частных сетей). По причине ненадёжности от telnet как средства управления операционными системами давно отказались.

Иногда клиенты telnet используются для доступа к другим символьно-ориентированным протоколам на основе транспорта TCP (HTTP, IRC, SMTP, POP3 и прочим). Однако, использование клиента telnet в этом качестве может привести к ошибкам, связанным с неправильной трактовкой сервером опций протокола

rlogin

Протокол rlogin (англ. remote login — удаленный вход в систему, RFC 1282, порт сервера – 513) позволяет пользователям рабочих станций UNIX подключаться к удаленным серверам UNIX через сеть Internet и работать так же, как при прямом подключении терминала к машине. Помимо rlogin существуют еще несколько подобных протоколов: rsh (remote shell), rcp (remote copy). Утилиты rlogin, rsh и rcp часто объединяют под общим названием r-команд.

Впервые появившаяся в составе 4.2BSD UNIX, программа rlogin (и остальные r-команды) одно время была исключительно популярной в этой среде. В качестве средства терминального доступа rlogin очень похожа на telnet, но из-за тесной интеграции с ОС нашла ограниченное применение в других системах.

В rlogin отсутствуют многие опции, свойственные telnet, в частности режим согласования параметров между клиентом и сервером. Однако rlogin предусматривает хотя бы минимальные средства защиты на основе доверительных отношений между хостами: на сервере rlogin в специальных системных файлах (обычно /etc/hosts.equiv и $HOME/.rhosts) администратор может перечислить компьютеры, доступ с которых к данному серверу будет разрешен без пароля.

Пользователи других компьютеров (не перечисленных в этих файлах) могут войти на сервер лишь после ввода пароля (который как и в telnet передается в открытом виде). Но, доверительные отношения между хостами тоже не панацея, и могут устанавливаться лишь в изолированных сетях. Дело в том, что такие техники несанкционированного доступа, как подмена IP-адресов (IP-spoofing) и доменных имен (DNS-spoofing), делают r-сервисы в Интернет незащищенными. Поэтому современные дистрибутивы UNIX-подобных ОС включают не фактические r-команды, а замещающие ссылки на их SSH-аналоги: scp, sftp и др.(см. man rlogin в Ubuntu, выводящий руководство по OpenSSH).

ПРОТОКОЛ SSH SSH (Secure Shell)

Первая версия протокола, SSH-1, была разработана в 1995 году исследователем Тату Улёненом из Технологического университета Хельсинки, Финляндия. SSH-1 был написан для обеспечения большей конфиденциальности, чем протоколы rlogin, telnet и rsh. В 1996 году была разработана более безопасная версия протокола, SSH-2, несовместимая с SSH-1. Протокол приобрел ещё большую популярность, и к 2000 году у него было около двух миллионов пользователей. В настоящее время под термином «SSH» обычно

подразумевается именно SSH-2, т.к. первая версия протокола ввиду существенных недостатков сейчас практически не применяется. В 2006 году протокол был утвержден рабочей группой IETF в качестве Интернет-стандарта.

Распространены две реализации SSH: собственническая коммерческая и бесплатная свободная. Свободная реализация называется OpenSSH. К 2006 году 80 % компьютеров сети Интернет использовало именно OpenSSH. Собственническая реализация разрабатывается организацией SSH Inc., она бесплатна для некоммерческого использования. Эти реализации содержат практически одинаковый набор команд.

Протокол SSH-2, в отличие от протокола telnet, устойчив к атакам прослушивания трафика («снифинг»), но неустойчив к атакам «человек посередине». Протокол SSH-2 также устойчив к атакам путем присоединения посредине (англ. session hijacking) - невозможно включиться в или перехватить уже установленную сессию.

Для предотвращения атак «человек посередине» при подключении к хосту, ключ которого ещё не известен клиенту, клиентское ПО показывает пользователю "слепок ключа" (key fingerprint). Рекомендуется тщательно проверять показываемый клиентским ПО "слепок ключа" (key fingerprint) со слепком ключа сервера, желательно полученным по надежным каналам связи или лично.

Поддержка SSH реализована во всех UNIX-подобных системах, и на большинстве из них в числе стандартных утилит присутствуют клиент и сервер ssh. Существует множество реализаций SSH-клиентов и для не-UNIX ОС. Большую популярность протокол получил после широкого развития анализаторов трафика и способов нарушения работы локальных сетей, как альтернативное небезопасному протоколу Telnet решение для управления важными узлами.

Для работы по SSH нужен SSH-сервер и SSH-клиент. Сервер прослушивает соединения от клиентских машин и при установлении связи производит аутентификацию, после чего начинает обслуживание клиента. Клиент используется для входа на удаленную машину и выполнения команд.

Для соединения сервер и клиент должны создать пары ключей — открытых и закрытых — и обменяться открытыми ключами. Обычно используется также и пароль.

Основными реализациями протоколами SSH являются:
SSH-серверы
BSD: OpenSSH
Linux: dropbear, lsh-server, openssh-server, ssh
Windows: freeSSHd, copssh, WinSSHD, KpyM Telnet/SSH Server, MobaSSH, OpenSSH через Cygwin
SSH-клиенты и оболочки
GNU/Linux, *BSD: kdessh, lsh-client, openssh-client, putty, ssh
MS Windows и Windows NT: PuTTY, SecureCRT, ShellGuard, Axessh, ZOC, SSHWindows, ProSSHD, XShell
MS Windows Mobile: PocketPuTTy, mToken, sshCE, PocketTTY, OpenSSH, PocketConsole
Mac OS: NiftyTelnet SSH
Symbian OS: PuTTY
Java: MindTerm, AppGate Security Server
J2ME: MidpSSH
iPhone: i-SSH, ssh (в комплекте с Terminal)
Android: connectBot
Blackberry: BBSSH
MAEMO 5: OpenSSH

SSH (англ. Secure SHell – безопасная оболочка) — безопасный протокол дистанционного управления компьютером. Основная спецификация – RFC 4251. The Secure Shell (SSH) Protocol Architecture, входящий порт – 22. По сравнению с протоколами telnet и rlogin, ssh отличается, в первую

очередь, высокой защищенностью за счет поддержки криптостойких алгоритмов шифрования и рядом дополнительных возможностей.

SSH позволяет не только использовать защищенную оболочку на удаленном хосте, но и туннелировать графический интерфейс (X11 forwarding), превращая клиентов, использующих X-Window System, в графические терминалы. SSH также способен (при надлежащем конфигурированиии) передавать через безопасный канал любой другой сетевой протокол – port forwarding. Криптографическая защита протокола SSH не фиксирована, возможен выбор различных алгоритмов шифрования.

Согласно Указу Президента Российской Федерации № 334 от 03.04.95 физическим лицам и организациям, включая государственные, частные и акционерные, запрещена эксплуатация систем криптографии, не прошедших сертификации в ФАПСИ. А SSH (а иже с ним PGP) является именно такой системой.

Не стоит думать, что нам пытаются запретить защищать конфиденциальную информацию: организации не только могут, но и обязаны защищать важную информацию. Только для этого они должны применять сертифицированные средства. Применяя свободное, но несертифицированное ПО на основе ssh, SSL, PGP и т.п. следует помнить, что его использование чревато разбирательством со стороны спецслужб.

Сервис SSH использует по умолчанию порт 22 и объединяет протоколы трех уровней:

- **Протокол аутентификации (The Secure Shell (SSH) Authentication Protocol, RFC 4252)** — этот уровень предоставляет несколько механизмов для аутентификации пользователя: здесь может использоваться традиционная парольная аутентификация, аутентификация, основанная на публичном ключе и т.д.
- **Протокол транспортного уровня (The Secure Shell (SSH) Transport Layer Protocol, RFC 4253)** — обеспечивает взаимодействие алгоритма и обмен ключами. Обмен ключами позволяет аутентифицировать сервер и создать криптографически защищённое соединение: обеспечивая целостность, конфиденциальность и дополнительное сжатие.
- **Протокол соединения (The Secure Shell (SSH) Connection Protocol, RFC 4254)** — создает безопасный (шифруемый) канал, представляя его в виде нескольких логических каналов, которые используются для различных целей (различных видов служб).
- **Поддержка SSH реализована во всех UNIX системах** и на большинстве из них в типовую поставку включено как клиентское, так и серверное ПО (см. man ssh). Для не-UNIX систем имеются как платное, так и бесплатное клиентское ПО (SecureCRT, putty). Серверное ПО – платное.

X-Window System

Практически все задачи управления UNIX могут выполняться в текстовом режиме, однако администраторы нередко предпочитают графический интерфейс, как более удобный. Вдобавок, некоторыми приложениями UNIX можно управлять только в графической среде. Программное обеспечение X-server, отвечающее за вывод графической информации, имеется для множества платформ, включая DOS, Windows, Macintosh, UNIX и т.д.

Следует иметь в виду, что применение X Window System предполагает наличие достаточно большой пропускной способности сети. Система прекрасно работает в локальных сетях, но очень медленно — по глобальным каналам. Поэтому при использовании X Window System на домашнем компьютере администратора управление лучше осуществлять через терминальные утилиты наподобие xterm, а не посредством графических утилит.

При подключении к серверу UNIX (на котором запускаются клиенты X11) аутентификация может осуществляться двумя методами: через терминальные утилиты (telnet, rlogin и т. п.) и через менеджер дисплеев X (X Display Manager, xdm). В первом варианте передачи пароля в открытом виде можно избежать, применяя вместо telnet и rlogin уже упоминавшиеся программы ssh и OTP. В случае X Display Manager пароли по умолчанию передаются в открытом виде. Поэтому при удаленном управлении сервером UNIX по общедоступным сетям xdm пользоваться не стоит.

Очень осторожно администраторы должны подходить к вопросу использования сервера UNIX в качестве сервера X (т. е., говоря понятным языком, к запуску графической оболочки X11 на сервере UNIX). X Window System устроена так, что пользователь может со своей машины запустить клиента X на удаленном сервере X и перехватывать на нем ввод/вывод информации. В результате злоумышленник получает возможность считывать конфиденциальную информацию с сервера X, включая пароли, вводимые пользователем на сервере X (хотя эмулятор терминала xterm позволяет блокировать перехват пароля, этой возможностью редко кто пользуется).

На серверах X применяются две схемы аутентификации клиентов: по имени хоста и с помощью «магических плюшек» (MIT-MAGIC-COOKIE-1). При аутентификации по имени хоста на сервере X создаются системные файлы, где перечисляются хосты, откуда разрешено запускать клиентские программы X на данном сервере X. Но подобную защиту никак не назовешь достаточной, так как с помощью подмены IP-адресов или доменных имен злоумышленник может провести атаку на X11.

При использовании же схемы «магических плюшек» (их поддержка встроена в протокол XDMCP, на основе которого функционирует X Display Manager) аутентификация осуществляется на основании учетных записей пользователей. Чтобы иметь право запустить клиента на сервере X, пользователь в своем домашнем каталоге машины-клиента X11 должен иметь системный файл с записанным секретным кодом сервера X. Этот секретный код и называется магической плюшкой. Беда только в том, что плюшка передается по сети в открытом виде, поэтому данный метод также вряд ли можно считать безопасным.

В X Window System 11 Release 5 добавлены еще две схемы (XDM-AUTHORIZATION-1 и SUN-DES-1), напоминающие схему MIT-MAGIC-COOKIE-1, но использующие алгоритм шифрования DES. Однако из-за экспортных ограничений такие схемы в комплект поставки X Window System не включают. Исходя из вышеприведенных соображений, запускать серверное ПО X11 на сервере UNIX можно лишь в том случае, когда запрещен доступ клиентов X11 с других компьютеров.

ПРОТОКОЛ RDP

Протокол удаленного рабочего стола RDP (Remote Desktop Protocol) обеспечивает удаленный доступ через сеть к рабочему столу компьютеров под управлением операционной системы Microsoft Windows. Используется при подключении тонких клиентов к терминальному серверу Windows с запущенной службой Microsoft Terminal Services. Разработан компанией Microsoft.

Ключевые возможности протокола RDP

Поддержка шифрования по алгоритму RC-4 с длиной ключа 128 или 56 бит

Поддержка протоколов TLS (Transport Layer Security)

Аутентификация пользователей с помощью смарт-карт (на сервере через службу терминальных подключений Microsoft Terminal Services)

Поддержка звука на локальном компьютере для приложений терминального сервера

File System Redirection – позволяет работать с файлами локального компьютера на удаленном терминальном сервере

Printer Redirection – позволяет печатать на принтере локального компьютера из приложений запущенных на удаленном терминальном сервере

Port Redirection – открывает доступ к последовательным и параллельным портам локального компьютера для приложений запущенных на удаленном терминальном сервере

Совместное использование буфера обмена как на локальном компьютере, так и на удаленном терминальном сервере

Глубина цвета дисплея: 24, 16, 15 или 8 бит

Невзирая на то, что сами пакеты проткола RDP передаются по сети в зашифрованном виде, сама терминальная сессия может быть подвергнута атаке Man In The Middle, так как ни серверная часть, ни клиентская не производят взаимную аутентификацию передаваемых и принимаемых пакетов с данными. Поэтому для построения полностью защищенных решений необходимо использовать защиту RDP на уровне SSL появившуюся в Windows Server 2003 Service Pack 1.

История версий

Первая версия RDP появилась в Terminal Services Windows NT 4.0. Основан на ITU-T T.128 application sharing protocol (проект, также известный как T.share) из серии рекомендаций T.120. Первая версия RDP была введена Microsoft в Terminal Services как часть их продукта Windows NT 4.0 Server, Terminal Server Edition. Она основывалась на MultiWin технологии Citrix, изначально поставлявшейся как часть Citrix WinFrame для Windows NT 3.51. Поддерживались несколько пользователей и сессий входа одновременно.[1]

Версия 5.0, появившаяся в Windows Server 2000, имеет дополнительные возможности, например, печать на локальные принтеры, и лучше использует пропускную способность сети.

Версия 5.1, появившаяся в Windows XP Professional, включала поддержку 24-битного цвета и звука.

Версия 5.2, появившаяся с Windows Server 2003, включает поддержку консоли, каталог сеанса и подключение (mapping) локальных ресурсов. Начиная с этой версии поддерживается проверка подлинности сервера по сертификату SSL и шифрование соединения по протоколу TLS 1.0.

Версия 6.0 установлена в Windows Vista и включила поддержку программ удаленного взаимодействия, приложениям Windows Presentation Foundation, поддержку нескольких мониторов

Версия 6.1 была выпущена в феврале 2007 и включена в Windows Server 2008, и в пакет обновления Windows Vista SP1 и Windows XP SP3.

В дополнение к изменениям, связанным с улучшенным доступом к консоли, эта версия включает новые функциональные возможности, появившиеся в Windows Server 2008, такие как Terminal Services Easy Print driver (новая клиентская система перенаправления принтера, которая позволяет выполнять локальную печать из приложений, выполняющихся на сервере, не устанавливая драйвер печати на сервере).

Версия 7 (вышла в составе Windows 7, поддерживается в Windows XP)

Поддержка аутентификации сетевого уровня— снижает риск успешной атаки типа DoS (Denial of Service)

Увеличение производительности ядра RDP

Поддержка технологии Windows Aero (Aero over Remote Desktop)

Поддержка технологий Direct2D и Direct3D 10.1 в приложениях

Полноценная поддержка мультидисплейных конфигураций

Улучшения в работе с мультимедиа

Поддержка технологии Media Foundation

Поддержка технологии DirectShow

Снижена длительность задержки при воспроизведении аудио

Версия 7.1 (вышла в составе Windows 7 SP1)

Версия 8.0 (вышла в составе Windows 8, с октября 2012 года доступна как пакет обновления для Windows 7 SP1 и Windows Server 2008)

Новые возможности появившиеся в шестой версии RDP

Remote Applications. Прямой запуск приложений на сервере в выделенной терминальной сессии без открытия окна терминальной сессии. Поддержка файловых ассоциаций локального компьютера – возможность запуска приложений на сервере для открытия документа на локальном компьютере в соотвествии с расширением в имени файла.

Seamless Windows. Эмуляция окна локального компьютера с запуском приложения на терминальном сервере. Автоматическая аутентификация на сервере с данными учетной записи пользователя. Автоматическое завершение соответствующей терминальной сессии при завершении работы приложения.

Terminal Server Gateway. Поддержка подключений RDP через сервер-шлюз IIS с использованием протокола https. Обеспечивает защищенное подключение к терминальному серверу расположенному за ISS в локальной сети предприятия.

Windows Aero Glass. Поддержка Windows Aero Glass включая сглаживание шрифтов ClearType.

Windows Presentation Foundation. Поддерживается на любых клиентах с установленной .NET Framework 3.0.

Полностью настраиваемые терминальные сервисы включая поддержку скриптов средствами Windows Management Instrumentation.

Улучшенное управление полосой пропускания для клиентов RDP.

Поддержка нескольких мониторов. Разделение экрана терминальной сессии на несколько мониторов. Работает только с системами Windows Vista.

Глубина цвета дисплея: 32, 24, 16, 15 или 8 бит

Благодаря использованию RDP версии 8.0 и RemoteFX пользователи виртуальных машин под управлением Windows 8, развернутых на узлах Windows Server 2012, получили несколько большие возможности по сравнению с использованием VDI (Virtual Desktop Infrastructure) операционных систем предыдущих поколений.

В частности, одной из особенностей VDI Windows 8 является предоставление широких возможностей виртуального рабочего стола при использовании не только на LAN-, но и на WAN-каналах. С выходом обновления протокола RDP 8.0 стало возможным использование виртуальных машин под управлением Windows 7 SP1 на хостах Windows Server 2012 в сценариях VDI со всеми усовершенствованиями. В частности, будут реализованы следующие возможности.

Для клиентских устройств доступа под управлением Windows 7 SP1 и Windows Server 2008 R2 SP1:

динамическое пренаправление USB-устройств — позволит пользователям выбирать необходимые устройства уже в процессе сессии удаленного рабочего стола,

улучшения в плане единой точки входа при веб-доступе к удаленным рабочим столам — предполагает под собой единоразовый ввод паролей для дальнейший соединений,

переподключение к удаленным рабочим столам и приложениям RemoteApp,

поддержка Lync 2013 в сценариях VDI — использование видео-конференций посредством клиента Lync в сессиях удаленных виртуальных рабочих столов.

Для виртуальных машин под управлением Windows 7 SP1:

RemoteFX для WAN-каналов — заявлена возможность использования RemoteFX на WAN-каналах путем оптимизации UDP-транспорта, автоопределение сетей при использовании RemoteFX — позволяет автоматически определить пропускую способность канала, адаптируя тем самым задержки передаваемые данные в условиях изменяющейся загрузки сети, адаптивная графика при использовании RemoteFX — предоставление возможностей графики различной сложности в зависимости от нагрузки сервера, клиента и сети, потоковые медиа-данные — возможности использования сглаживаемых показателей передачи мультимедиа-контента, перенаправление USB в RemoteFX для виртуальных рабочих столов без виртуальной видеокарты, поддержка вложенных сессий — запуск “RDP в RDP” официально признан поддерживаемым, счетчики производительности для мониторинга пользователей.

Remote Desktop Protocol

RDP (англ. Remote Desktop Protocol, протокол удалённого рабочего стола) — открытый протокол прикладного уровня, разработанный Microsoft и изначально предназначенный для подключения графических терминалов к Windows Terminal Server. Клиенты существуют практически для всех версий Windows (включая Windows CE и Mobile), Linux, Free BSD, Mac OS X. По-умолчанию используется порт TCP 3389. Официальное название Майкрософт для клиентского ПО – Remote Desktop Connection или Terminal Services Client (TSC).

RDP-клиент, – Remote Desktop – позволяет вам, находясь на одном компьютере, удаленно управлять другим. Например, если вам нужно зайти в свой компьютер, находящийся в офисе, из дома (предположим, что вы заблаговременно настроили свой рабочий компьютер), то вы можете с помощью инструмента Remote Desktop получить доступ ко всем данным, находящимся на рабочем компьютере, включая файлы, приложения и сетевые соединения.

Фактически Remote Desktop позволяет не только получать доступ к файлам удаленного компьютера, но и на самом деле видеть рабочий стол таким, какой он есть на удаленном компьютере. Более того, если удаленный компьютер работает в операционной системе начиная с Windows 2000 или .NET Server, то на нем могут удаленно работать несколько пользователей одновременно, но для этого необходима терминальная лицензия.

Технология **Terminal Services** является основой для работы удаленного помощника , который позволяет вашему другу или работнику технической поддержки устанавливать соединение с вашим компьютером, видеть ваш рабочий стол и управлять компьютером.

Для соединения инструмент Remote Desktop использует LAN, виртуальную частную сеть (VPN) или интернет-соединение. Работа удаленного рабочего стола сильно зависит от скорости установленного соединения.

Remote Desktop поддерживает работу в двух режимах:

- Remote Desktop (Удаленный рабочий стол) – подходит для использования в локальной сети и требует установки программного обеспечения на компьютере-клиенте.
- Remote Desktop Web Connection (Интернет-подключение к удаленному рабочему столу) – требует на клиентской машине только наличия браузера Internet Explorer, но на сервере для нее необходимо установить и настроить большее количество программ.

Remote Desktop поддерживает 24-битные цвета – это позволяет варьировать качество картинки в широких пределах.

Переадресация ресурсов позволяет, например, использовать файловую систему удаленного компьютера в качестве сетевого ресурса общего доступа.

Переадресация звуков позволяет компьютеру-клиенту воспроизводить звуки, которые генерируются на компьютере-сервере. При проигрывании звуков Remote Desktop также учитывает пропускную способность полосы частот. Вместо того чтобы перегружать соединение звуковым сигналом при изменении пропускной способности, Remote Desktop снижает качество звука.

Remote Desktop-сервер и Remote Desktop-клиент пользуются общим буфером. Это позволяет им свободно обмениваться информацией.

Веб-технологии удаленного управления

Всемирная Паутина (World Wide Web) оказывает все большее влияние на средства управления сетевой средой. Уже сейчас многие маршрутизаторы, коммутаторы, сетевые принтеры допускают управление через web-браузеры.

Но список этот далеко не исчерпывается ими и Web вторгается и в сферу управления сетевыми ОС.

Вначале из Web можно было управлять лишь серверами HTTP и FTP, но этот список постоянно расширяется и охватывает теперь СУБД (например, phpMyAdmin как веб-интерфейс к СУБД MySQL), файловые системы, межсетевые экраны, сетевые службы DNS, DHCP и многое другое (пример – Webmin для управления UNIX-подобными системами).

Несмотря на все увеличивающееся количество подобных решений, до полнофункциональных специализированных административных систем они не доросли. Основная проблема здесь в том, что для многих приложений и, особенно, сетевых устройств пароль по HTTP передается в открытом виде.

Протокол PPTP

Поддержка PPP позволяет компьютерам с Windows Server получать с помощью подключений удаленного доступа к сети доступ к любому серверу, также поддерживающему стандарт PPP. Совместимость со стандартом PPP позволяет компьютеру с Windows Server принимать входящие вызовы и предоставлять доступ к сети клиентам удаленного доступа, использующим программное обеспечение других разработчиков.

Архитектура протокола PPP также позволяет клиентам удаленного доступа использовать любую комбинацию протоколов IPX, TCP/IP, NetBEUI и AppleTalk. Клиенты удаленного доступа операционных систем Windows могут использовать любую комбинацию протоколов TCP/IP, IPX и NetBEUI и программ с интерфейсом Windows Sockets, NetBIOS и IPX.

PPTP (Point-to-Point Tunneling Protocol) — туннельный протокол типа точка-точка, позволяющий компьютеру устанавливать защищённое соединение с сервером за счёт создания специального туннеля в стандартной, незащищённой сети. PPTP помещает (инкапсулирует) кадры PPP в IP-пакеты для передачи по глобальной IP-сети, например Интернет. PPTP может также использоваться для организации туннеля между двумя локальными сетями. PPTP использует дополнительное TCP-соединение для обслуживания туннеля.

Спецификация протокола была опубликована как «информационная» RFC 2637 в 1999 году. Она не была ратифицирована IETF. Протокол считается менее безопасным, чем IPSec. PPTP работает, устанавливая обычную PPP сессию с противоположной стороной с помощью протокола Generic Routing Encapsulation. Второе соединение на TCP-порте 1723 используется для инициации и управления GRE-соединением. PPTP сложно перенаправлять за сетевой экран, так как он требует одновременного установления двух сетевых сессий.

PPTP-трафик может быть зашифрован с помощью MPPE. Для аутентификации клиентов могут использоваться различные механизмы, наиболее безопасные из них — **MS-CHAPv2 и EAP-TLS.**

Компания Cisco первой реализовала PPTP и позже лицензировала эту технологию корпорации Microsoft. PPTP удалось добиться популярности благодаря тому, что это первый протокол туннелирования, который был поддержан корпорацией Microsoft. Все версии Microsoft Windows, начиная с Windows 95 OSR2, включают в свой состав PPTP-клиент, однако существует ограничение на два одновременных исходящих соединения. А сервис удалённого доступа для Microsoft Windows включает в себя PPTP сервер.

Некоторое время в Linux-дистрибутивах отсутствовала полная поддержка PPTP из-за опасения патентных претензий по поводу протокола MPPE. Впервые полная поддержка MPPE появилась в Linux 2.6.13 (2005 год).

Официально поддержка PPTP была начата с версии ядра Linux 2.6.14. Тем не менее, сам факт применения MPPE в PPTP фактически не обеспечивает безопасность протокола PPTP.

Операционная система FreeBSD поддерживает PPTP протокол, используя в качестве сервера PPTP порт mpd (/usr/ports/net/mpd), используя подсистему netgraph; можно также использовать программу PoPToP (/usr/ports/net/poptop). В качестве клиента PPTP в системе FreeBSD может выступать либо порт pptpclient (/usr/ports/net/pptpclient), либо порт mpd, работающий в режиме клиента.

Mac OS X поставляется со встроенным PPTP клиентом. Cisco и Efficient Networks продают реализации PPTP клиента для более старых версий Mac OS. КПК Palm, имеющие поддержку Wi-Fi, поставляются с PPTP клиентом Mergic.

Microsoft Windows Mobile также поддерживают PPTP.

PPTP был объектом множества анализов безопасности, в нём были обнаружены различные серьёзные уязвимости. Известные относятся к используемым протоколам аутентификации PPP, устройству протокола MPPE, и интеграции между аутентификациями MPPE и PPP для установки сессионного ключа. Краткий обзор данных уязвимостей:

MSCHAP-v1 совершенно ненадёжен. Существуют утилиты для лёгкого извлечения хешей паролей из перехваченного обмена MSCHAP-v1.

MSCHAP-v2 уязвим к словарной атаке на перехваченные challenge response пакеты. Существуют программы, выполняющие данный процесс.

В 2012 году было показано, что сложность подбора ключа MSCHAP-v2 эквивалентна подбору ключа к шифрованию DES, и был представлен онлайн-сервис, который способен восстановить ключ за 23 часа.

При использовании MSCHAP-v1, MPPE использует одинаковый RC4 сессионный ключ для шифрования информационного потока в обоих направлениях. Поэтому стандартным методом является выполнение XOR'а потоков из разных направлений вместе, благодаря чему криптоаналитик может узнать ключ.

MPPE использует **RC4** поток для шифрования. Не существует метода для аутентификации цифробуквенного потока и поэтому данный поток уязвим к атаке, делающей подмену битов. Злоумышленник легко может изменить поток при передаче и заменить некоторые биты, чтобы изменить исходящий поток без опасности своего обнаружения. Данная подмена битов может быть обнаружена с помощью протоколов, считающих контрольные суммы.

ПРОТОКОЛ SLIP

SLIP (Serial Line Internet Protocol) — устаревший сетевой протокол канального уровня эталонной сетевой модели OSI для доступа к сетям стека TCP/IP через низкоскоростные линии связи путём простой инкапсуляции IP-пакетов. Используются коммутируемые соединения через последовательные порты для соединений клиент-сервер типа точка-точка. В настоящее время вместо него используют более совершенный протокол PPP.

Протокол SLIP (Serial Line Internet Protocol) является старым стандартом удаленного доступа, используемым, в основном, серверами удаленного доступа на платформе UNIX. Клиенты удаленного доступа, использующие Windows Server, поддерживают SLIP и могут подключаться к любому серверу удаленного доступа, также поддерживающему стандарт SLIP. Это позволяет клиентам Windows NT версии 3.5 и последующих версий подключаться к множеству существующих UNIX-серверов.SLIP был разработан в начале 80-х компанией 3COM. Протокол начал быстро распространяться после включения в ОС Berkeley Unix 4.2 Риком Адамсом (Rick Adams) в 1984, так как благодаря ему стало возможным подключение к Интернет через последовательный COM-порт, имевшийся на большинстве компьютеров. Ввиду своей

простоты сейчас используется в микроконтроллерах
Недостатки

- Нет возможности обмениваться адресной информацией — необходимость предустановки IP-адресов.
- Отсутствие индикации типа инкапсулируемого протокола — возможно использование только IP.
- Не предусмотрена коррекция ошибок — необходимо выполнять на верхних уровнях, рекомендуется использовать протокол TCP.
- Высокая избыточность — из-за использования стартовых и стоповых битов при асинхронной передаче (+20 %), передачи в каждом SLIP-кадре полного IP-заголовка (+20 байт) и полных заголовков верхних уровней, байт-стаффинга.
- В некоторых реализациях протокола максимальный размер кадра ограничен 1006 байтами для достижения обратной совместимости с реализацией в Berkeley Unix.

ПРОТОКОЛ PPP

PPP (англ. Point-to-Point Protocol) — двухточечный протокол канального уровня сетевой модели OSI. Обычно используется для установления прямой связи между двумя узлами сети, причем он может обеспечить аутентификацию соединения, шифрование (с использованием ECP, RFC 1968) и сжатие данных. Используется на многих типах физических сетей: нуль-модемный кабель, телефонная линия, сотовая связь и т. д.

Часто встречаются подвиды протокола PPP, такие как Point-to-Point Protocol over Ethernet (PPPoE), используемый для подключения по Ethernet, и иногда через DSL; и Point-to-Point Protocol over ATM (PPPoA), который используется для подключения по ATM Adaptation Layer 5 (AAL5), который является основной альтернативой PPPoE для DSL.

PPP представляет собой целое семейство протоколов: протокол управления линией связи (LCP), протокол управления сетью (NCP), протоколы аутентификации (PAP, CHAP), многоканальный протокол PPP (MLPPP).

PPP протокол был разработан на основе HDLC (High-Level Data Link Control- бит-ориентированный протокол канального уровня сетевой модели OSI, разработанный ISO) и дополнен некоторыми возможностями, которые до этого встречались только в проприетарных протоколах.

Link Control Protocol (LCP) обеспечивает автоматическую настройку интерфейсов на каждом конце (например, установка размера пакетов) и опционально проводит аутентификацию. Протокол LCP работает поверх PPP, то есть начальная PPP связь должна быть до работы LCP.

RFC 1994 описывает Challenge-handshake authentication protocol (CHAP), который является предпочтительным для соединений с провайдерами. Уже устаревший Password authentication protocol (PAP) всё еще иногда используется.Другим вариантом аутентификации через PPP является Extensible Authentication Protocol (EAP).

После того, как соединение было установлено, поверх него может быть настроена дополнительная сеть. Обычно используется Internet Protocol Control Protocol (IPCP), хотя Internetwork Packet Exchange Control Protocol (IPXCP) и AppleTalk Control Protocol (ATCP) были когда-то популярны. Internet Protocol Version 6 Control Protocol (IPv6CP) получит большее распространение в будущем, когда IPv6 заменит IPv4 как основной протокол сетевого уровня.

PPP позволяет работать нескольким протоколам сетевого уровня на одном канале связи. Другими словами, внутри одного PPP-соединения могут передаваться потоки данных различных сетевых протоколов (IP, Novell IPX и т. д.), а также данные протоколов канального уровня локальной сети. Для каждого сетевого протокола

используется Network Control Protocol (NCP) который его конфигурирует (согласовывает некоторые параметры протокола).

PPP обнаруживает закольцованные связи, используя особенность, включающую magic numbers. Когда узел отправляет PPP LCP сообщения, они могут включать в себя магическое число. Если линия закольцована, узел получает сообщение LCP с его собственным магическим числом вместо получения сообщения с магическим числом клиента. Link Control Protocol устанавливает и завершает соединения, позволяя узлам определять настройки соединения. Также он поддерживает и байто-, и бито-ориентированные кодировки. Network Control Protocol используется для определения настроек сетевого уровня, таких как сетевой адрес или настройки сжатия, после того как соединение было установлено.

Так как в PPP входит LCP протокол, то можно управлять следующими LCP параметрами:

- Аутентификация. RFC 1994 описывает Challenge Handshake Authentication Protocol (CHAP), который является предпочтительным для проведения аутентификации в PPP, хотя Password Authentication Protocol (PAP) иногда еще используется. Другим вариантом для аутентификации является Extensible Authentication Protocol (EAP).

- Сжатие. Эффективно увеличивает пропускную способность PPP соединения, за счет сжатия данных в кадре. Наиболее известными алгоритмами сжатия PPP кадров являются Stacker и Predictor.

- Обнаружение ошибок. Включает Quality-Protocol и помогает выявить петли обратной связи посредством Magic Numbers RFC 1661.

- Многоканальность. Multilink PPP (MLPPP, MPPP, MLP) предоставляет методы для распространения трафика через несколько физических каналов, имея одно логическое соединение. Этот вариант позволяет расширить пропускную способность и обеспечивает балансировку нагрузки.