4. СЕТЕВЫЕ ВОЗМОЖНОСТИ UNIX

4.1 Служба DNS

В основе сетевых возможностей UNIX лежит стек протоколов TCP/IP, образующих основу Интернет. Протокол IP обеспечивает однозначную идентификацию сетевого интерфейса в глобальной сети. Значение IP-адреса состоит из четырех чисел в пределах от 0 до 255, разделённых точками (например, 192.168.12.134). Чаще всего компьютер (хост в сети Интернет) имеет всего один сетевой интерфейс, и поэтому говорят про IP-адрес хоста. Кроме IP-адресов используются имена хостов (доменные имена). Для большинства хостов установлено специальное отображение между именем хоста и его IP-адресом. Такое соответствие возможно благодаря работе DNS (Domain Name Service – служба имён доменов). Если DNS содержит запись о том, что *www.domain.com* соответствует 192.168.42.7, то при обращении к сервисам этого хоста можно использовать данное доменное имя.

Для получения информации из DNS используются команды **dig** и **host**. Эти команды имеют множество параметров. В самом простейшем случае в качестве параметра указывается имя хоста, DNS-запись о котором вы хотите получить. Например, получить DNS-запись о хосте *ya.ru*:

```
$ dig ya.ru
 ; <<>> DiG 9.2.4 <<>> ya.ru
 ;; global options:  printcmd
 ;; Got answer:
 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 453
 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2
 ;; QUESTION SECTION:
 ;ya.ru.                 IN  A
 ;; ANSWER SECTION:
 ya.ru.             306  IN  A   213.180.204.8
 ;; AUTHORITY SECTION:
 ya.ru.             306  IN  NS  ns1.yandex.ru.
 ya.ru.             306  IN  NS  ns5.yandex.ru.
 ;; Query time: 5 msec
 ;; SERVER: 194.85.172.133#53(ns.karelia.ru)
 ;; WHEN: Sun Jan 28 23:47:04 2018
 ;; MSG SIZE  rcvd: 82
```

Ответ команды **dig** состоит из нескольких секций. Сначала идёт секция с общей информацией. Затем идёт секция, описывающая запрос к службе DNS (QUERY SECTION). Третья секция (ANSWER SECTION), собственно, и содержит требуемую информацию. Наибольший интерес представляют первая колонка (доменное имя) и последняя колонка (IP-адрес). В следующей секции (AUTHORITY SECTION) сообщается, какие DNS-хосты предоставили эту информацию. В отчёте программы может присутствовать секция с дополнительными DNS-записями, касающимися данного запроса.

По умолчанию результат работы команды **host** не содержит такого обилия информации:

```
$ host ya.ru
 ya.ru has address 213.180.204.8
```

С помощью **dig** и **host** можно выполнять обратный поиск (reverse mapping) – определение доменного имени по IP-адресу.

```
$ dig -x 213.180.204.8
 ; <<>> DiG 9.2.4 <<>> -x 213.180.204.8
 ;; global options:  printcmd
 ;; Got answer:
 ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 197
 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2
 ;; QUESTION SECTION:
 ;8.204.180.213.in-addr.arpa.      IN  PTR
 ;; ANSWER SECTION:
 8.204.180.213.in-addr.arpa. 14384 IN  PTR  ya.ru.
 ;; AUTHORITY SECTION:
 204.180.213.in-addr.arpa.   85632 IN NS ns2.yandex.net.
 204.180.213.in-addr.arpa.   85632 IN NS ns1.yandex.net.
 ;; Query time: 4 msec
 ;; SERVER: 194.85.172.133#53(ns.karelia.ru)
```

```
 ;; WHEN: Wed Jan 31 23:56:56 2018
 ;; MSG SIZE  rcvd: 109 ;; MSG SIZE  rcvd: 91
$ host 213.180.204.8
 8.204.180.213.in-addr.arpa domain name pointer ya.ru.
```

4.2 Команда ping

С помощью команды **ping** можно проверить наличие соединения с тем или иным хостом. Работа команды состоит в том, что по указанному адресу посылаются небольшие порции данных (пакеты) и регистрируется их благополучный приход обратно. Если связи нет (линия отсутствует или система назначения выключена), то пакеты не возвращаются. Работа команды прерывается комбинацией клавиш [Ctrl]+[c]. При этом выдаётся статистика:

```
$ ping ns.karelia.ru
PING ns.karelia.ru(194.85.172.133) 56(84) bytes of data.
64 bytes from ns.karelia.ru (194.85.172.133): icmp_seq=0 ttl=62 time=16.5 ms
64 bytes from ns.karelia.ru (194.85.172.133): icmp_seq=1 ttl=62 time=8.72 ms
64 bytes from ns.karelia.ru (194.85.172.133): icmp_seq=2 ttl=62 time=20.2 ms
^C
--- ns.karelia.ru ping statistics ---
3 packets transmitted, 3 received, 0% loss, time 2004ms
rtt min/avg/max/mdev = 8.724/15.154/20.217/4.791 ms
```

Следует отметить, что не всякая система отвечает на запросы команды **ping**, что, тем не менее не является показателем её неработоспособности. Возможность ответа системы на запросы команды **ping** определяется её конфигурацией (например, настройками сетевого экрана – брэндмауэра).

4.3 Обмен сообщениями между пользователями

Обмен краткими сообщениями обеспечивает команда **write.** Текст сообщения берётся со стандартного ввода команды (для завершения ввода можно использовать [Ctrl]+[d]). Общий синтаксис:

write *имя_пользователя*

При использовании **write** адресат получает уведомление:

```
Write: Message from имя_пользователя@имя_хоста
```

после которого следует текст сообщения. О завершении текста сигнализирует появление «EOF». Возникающие сообщения могут смешаться с набираемыми пользователем командами и результатом их работы, поэтому опытные пользователи открывают одновременно несколько сеансов, один из которых используется только для обмена сообщениями. Кроме того, пользователь может заблокировать обмен сообщениями при помощи команды **mesg n**. В таком случае он не сможет ни сам отправить сообщения, ни другой пользователь не сможет побеспокоить его своими сообщениями. Отмена блокировки производится командой **mesg y**.

Возможность интерактивного режима обмена информацией предоставляет команда

talk *имя_пользователя@имя_хоста*

Эта команда выводит отправителя и адресата из режима командной строки. При вызове **talk** у отправителя экран делится на две части по горизонтали и в верхнем левом углу появляется надпись «Waiting for respond». Одновременно с этим у адресата на экране появляется сообщение:

```
talk: Message from talk daemon on имя_хоста ...
talk: Answer to имя_пользователя@имя_хоста ...
```

сопровождаемое звуковым сигналом. Если адресат не реагирует, то это сообщение будет появляться каждые 3 секунды до тех пор, пока отправитель не прекратит работу программы **talk** нажатием [Ctrl]+[c]. Если адресат хочет установить соединение, он должен при появлении вышеуказанного сообщения также вызвать **talk** с указанием абонента. При этом у обоих участников связи экран делится на две части по горизонтали и в верхнем левом углу появляется сообщение «Connection established», после чего оба участника могут набирать текст своих сообщений на экране одновременно. Для каждого из участников текст, набираемый им, отображается в верхней половине экране, а текст, набираемый его абонентом – в нижней половине. Выход из программы **talk** осуществляется по [Ctrl]+[c]. Если один из участников выходит из программы, другой видит в верхнем левом углу экрана сообщение «Connection closed. Exiting» и программа **talk** с его стороны прекращает работу.

Командами **write** и **talk** можно пользоваться, если существует уверенность, что те пользователи, с которыми есть необходимость пообщаться, в данный момент подключены к системе (см. команды **who** и **finger**).

4.4 Средства удалённого доступа

TELNET – это прикладной протокол, позволяющей любой машине работать в качестве удалённого *терминала* UNIX-хоста. Программа, реализующая этот протокол (обычно она называется **`telnet`**), посылает коды нажатых пользователем клавиш удалённой машине под управлением UNIX и выводит на экран символы, получаемые от удалённого хоста. Это позволяет войти на другую машину и работать на ней, как будто вы сидите за её системной консолью. В большинстве операционных систем синтаксис команды таков: **`telnet`** *`сервер`*. Чтобы завершить работу **`telnet`**, используются команды **`exit`** или **`logout`**.

В настоящее время протокол TELNET используется всё реже и реже. Это объясняется остро стоящими вопросами безопасности. Все данные передаются в открытом виде (рис. 4.3). Серверу передаются символы, соответствующие нажатым на клавиатуре клиента клавишам, а клиент получает от сервера символы для вывода на экран. Информация уязвима для перехвата. Если злоумышленник перехватывает сетевые пакеты, ему могут достаться пароли и другая персональная информация пользователей. На смену TELNET приходит протокол SSH, обеспечивающий шифрование передаваемых данных (рис. 4.4).

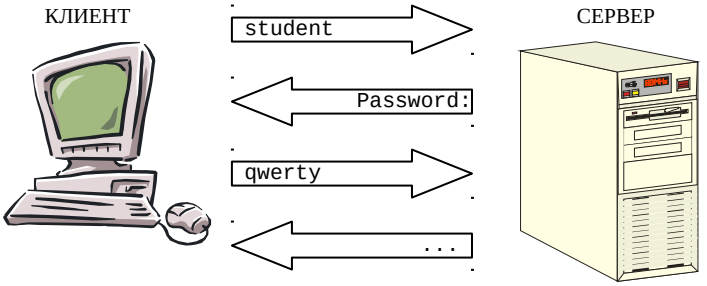

Рис. 4.3. Взаимодействие клиента и сервера по протоколу TELNET

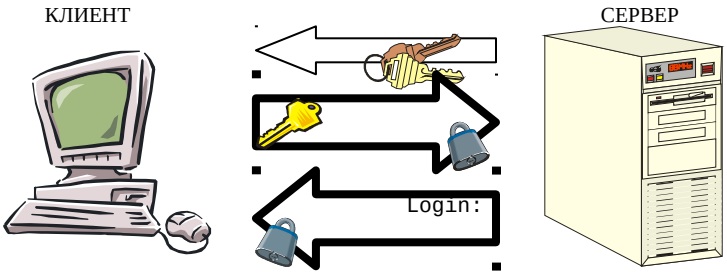

Рис. 4.4. Взаимодействие клиента и сервера по протоколу SSH

Работа SSH основана на использовании симметричных и асимметричных алгоритмов шифрования. Класс симметричных алгоритмов характеризуется тем, что для шифрования и расшифровки используется один и тот же ключ. Он секретен для стороннего наблюдателя, но его должны знать и отправитель, и получатель сообщения. Симметричные алгоритмы используются для быстрого шифрования. В асимметричном шифровании используется два ключа – один для шифрования, другой для расшифровки. В этом случае один из ключей является секретным, а другой общедоступным (открытым), поэтому такой класс алгоритмов ещё называют шифрованием с открытым ключом. При этом не должно существовать эффективной процедуры для получения секретного ключа по известному открытому ключу. В симметричной криптографии есть принципиально не раскрываемые шифры (так называемые теоретически стойкие или совершенно секретные системы Клода Шеннона), правда, для этого нужно, чтобы размер ключа был равен объёму текста. Асимметричные шифры раскрываемы в принципе, а знание секретного ключа лишь ускоряет дешифрование. Суть технологии асимметричного шифрования заключается в использовании такого алгоритма, вычисление которого со знанием некоторого секрета имело бы полиномиальную сложность ($\sim n^a$), а без этого знания – экспоненциальную ($\sim e^n$). Увеличивая число разрядов (n), можно добиться того, что решение задачи на современных вычислительных устройствах без знания секрета потребует нескольких сотен или даже тысяч лет, а знание секрета позволяет решить эту же задачу достаточно быстро. Асимметричные криптоалгоритмы обычно применяют к небольшим объёмам данных.

Схематично описать процедуру установления соединения по протоколу SSH можно следующим образом (рис. 4.4):

1. Сервер передаёт клиенту свой открытый ключ.
2. а) Клиент генерирует ключ сессии для симметричного криптоалгоритма.
   б) Клиент шифрует ключ сессии открытым ключом сервера по асимметричному криптоалгоритму (RSA) и пересылает это зашифрованное сообщение серверу.
   в) Сервер получает и расшифровывает ключ сессии при помощи своего секретного ключа.
3. Аутентификация пользователя (передача имени, пароля) и дальнейший обмен данными происходит в виде сообщений, зашифрованных при помощи ключа сессии по симметричному криптоалгоритму (AES, IDEA, 3DES, RC4 и др.)

4.5 Передача файлов

Протокол FTP (File Transfer Protocol) предназначен для передачи файлов в глобальной сети Интернет. Сами программы, реализующие данный протокол для различных систем, как правило, носят то же самое имя – **ftp**. Используя **ftp**, вы можете копировать файлы с удалённой системы (на профессиональном жаргоне – «скачивать») либо на удалённую систему (т. е. загружать их туда). Набор команд **ftp** в различных реализациях программы не совсем одинаков. Принципы сохранены, и основные команды одни, но некоторые тонкости в зависимости от системы всегда есть.

Для получения помощи по конкретной команде предназначена команда **help** или **?**. При использовании без аргумента выводится полный список всех команд.

Открыть соединение с хостом – команда **open** *адрес хоста*. Чтобы продолжить работу вам необходимо либо зарегистрироваться в системе под своим именем, либо использовать анонимное подключение. В первом варианте вы вводите имя своей учётной записи (login) и пароль. Во втором варианте вместо имени вы вводите «anonymous» (или «ftp»), а в качестве пароля – свой электронный адрес (некоторые системы разрешают в этом случае оставлять пароль пустым). Не все системы разрешают анонимное подключение.

Команды для просмотра содержимого и смены текущего каталога идентичны уже знакомым командам – **ls** и **cd** (в некоторых FTP-клиентах – **dir** и **cdup**, соответственно).

Перед загрузкой файлов следует обратить внимание на режим загрузки файлов. FTP различает два режима загрузки файлов: бинарный (image, binary) и текстовый (ascii). При использовании текстового режима передаваемые данные могут подвергаться перекодированию между текстовыми форматами разных систем. Например, в UNIX строки текстовых файлов завершаются только символом «перевод строки» (ASCII 10), а в MS-DOS и Windows строки завершаются парой символов: «возврат каретки» (ASCII 13) и «перевод строки» (ASCII 10). FTP позволяет автоматически выполнять требуемое преобразование. При передаче в бинарном режиме никаких преобразований не осуществляется – файл просто передаётся байт за байтом. Если попытаться передать в текстовом режиме файл нетекстового формата (например, архив, исполняемый файл и т. п.), он будет испорчен. Если возникают сомнения – используйте бинарный режим. Команды **binary** и **ascii** используются для перевода, соответственно, в бинарный режим передачи файлов и в текстовый.

Командой **hash** можно включить вывод меток. Метки выводятся на экран при передаче очередного блока информации и отражают ход передачи информации.

Ваш локальный каталог – это каталог вашей системы, куда вы хотите в конечном счёте сохранить файлы. В то время как команда **cd** меняет каталог удалённой машины (машины, на которую вы вошли по FTP), командой **lcd** можно сменить локальный каталог. Кроме **lcd** есть и другие команды для работы с локальным диском.

Процесс загрузки файла с удалённой системы на локальную осуществляется командой

get *имя_файла_на_удалённой_машине* [*локальное_имя*]

Чтобы загрузить файл на удалённую систему, используется команда **put**.

Для прекращения FTP-сессии используется команда quit или bye. Команда close может использоваться для закрытия связи с данным FTP-сервером, но без выхода из программы ftp. Затем команда open может быть использована для начала новой FTP-сессии.

Протокол FTP так же, как и TELNET, передаёт данные (в том числе при регистрации пользователя) в открытом виде. Если это неприемлемо, используется протокол SFTP (Secure FTP), реализованный как подсистема протокола SSH. Для работы с протоколом SFTP используются программы **sftp**, **scp** и др.

4.6 Электронная почта (E-mail)

Электронная почта (e-mail) – один из самых распространённых методов обмена информацией в сетевых системах. Принципы работы с электронной почтой для пользователя напоминают традиционную бумажную почту. Получив учётную запись в системе UNIX, зачастую вы автоматически получаете почтовый ящик с именем *имя@хост* и возможность обмена электронной почтой. Ваш электронный адрес складывается из вашего регистрационного имени и имени хоста. И то и другое нетрудно узнать посредством команд **whoami** и **hostname**. Каждый раз система, получая почту на ваш адрес, будет помещать её в ваш почтовый ящик. Почтовый ящик – это обыкновенный файл.

Основную роль в работе почты играет программа **sendmail**. Она осуществляет приём почты для всех пользователей системы, «раскладывает» почту по почтовым ящикам, а также принимает от вашего почтового клиента исходящую почту. Самые простые почтовые клиенты для UNIX – программы **mail** или **mailx**. Доступ к вашему почтовому ящику можно получить с удалённой машины при помощи протоколов POP3 или IMAP4. Наиболее популярные почтовые клиенты для Windows, реализующие эти протоколы: The Bat!, Mozilla Thunderbird, Outlook Express и др. В текстовом терминале UNIX обычно для этих целей используют программы **pine**, **elm** или **mutt**.

Контрольные вопросы и задания

1. Просмотрите справочное руководство по командам **dig** и **host**. Определите IP-адреса указанных преподавателем хостов. Выполните обратный поиск.
2. Посредством команд **write** и **talk** произведите обмен сообщениями с пользователями, работающими в сети.
3. Изучите работу команды **ftp**. Загрузите несколько небольших файлов с одного из анонимных FTP-серверов. По протоколу **sftp** осуществите обмен файлами с указанной преподавателем машиной.
4. Воспользуйтесь командой **ssh** и произведите подключение к указанному преподавателем удалённому серверу UNIX. По справочному руководству **ssh** изучите альтернативные способы аутентификации в системе (без пароля при помощи ключей асимметричных криптоалгоритмов – SSH version 2, the public key method). Настройте указанную преподавателем систему на такой способ аутентификации.
5. Воспользуйтесь любой почтовой программой (например, **pine**) для чтения собственной почты и её отправки. Воспользуйтесь справочным руководством и встроенной помощью.
6. Изучите работу текстовых web-браузеров **links** или **lynx**. Выясните, как в них задаётся прокси-сервер, как выбирается кодировка страниц.