Липецкий государственный технический университет

Кафедра прикладной математики

Отчет по лабораторной работе №7
«Работа с SSH. Авторизация по ключу SSH.»

Студент _____ Стукановский А.О.
подпись, дата фамилия, инициалы

Группа ПМ-18

Руководитель
доц., к.п.н. кафедры АСУ _____ Кургасов В. В.
ученая степень, ученое звание подпись, дата фамилия, инициалы

Липецк 2021 г.

Содержание

Цель работы

Лабораторная работа предназначена для целей практического ознакомления с программным обеспечением удалённого доступа к распределённым системам обработки данных.

Практическое задание

1) Создать подключение удаленного доступа к системе обработки данных, сформировать шифрованные ключи и произвести их обмен с удаленной системой, передать файл по шифрованному туннелю, воспользовавшись беспарольным доступом с аутентфикацией по публичным ключам.

2) Выполнить подключение с использованием полноэкранного консольного оконного менеджера screen.

3) Запустить терминал с командной оболочкой ОС и ввести команду tmux (терминальный мультиплексор). Комбинациями клавиш Ctrl-b с создать новое окно и запустить анализатор трафика tcpdump с фильтром пакетов получаемых и передаваемых от узла domen.name с TCP-портом источника и назначения 23. С помощью команды tee, вывести отфильтрованные IP-пакеты на терминал и сохранить данные в файл telnet.log, в домашнем каталоге пользователя. Для этого следует воспользоваться командой sudo tcpdump -l -v -nn tcp and src port 23 or dst port 23 | tee telnet.log;

4) В первом окне терминального мультиплексора попытаться установить соединение с удаленным сервером domen.name по протоколу TELNET. Для авторизации следует использовать логин student; /при возможности организовать такой доступ инженерами кафедры АСУ ЛГТУ/

5) Воспользовавшись окном сетевого монитора, анализировать прохождение сетевых пакетов между узлами назначения. Отметить пакеты инициации соединения telnet;

6) Подключившись к удаленной системе ввести пароль Password и выполнить команду uname -a, выведя тем самым информацию об удаленной системе. Для разрыва соединения использовать команду logout;

7) В окне сетевого монитора отметить пакеты инициирующие разрыв сессии telnet. Прервать фильтрацию пакетов сетевым анализатором tcpdump, воспользовавшись комбинацией Ctrl-c. В файле telnet.log выделить записи установления и разрыва соединения с сервером telnet;

8) Снова запустить анализатор сетевого трафика с фильтром пакетов получаемых и передаваемых узлу domen.name с TCP-портом источника и назначения 22. С помощью команды tee, вывести отфильтрованные IP-пакеты на терминал и сохранить данные в файл ssh.log, в домашнем каталоге пользователя. Для этого следует воспользоваться командой sudo tcpdump -l -v -nn tcp and src port 22 or dst port 22 | tee ssh.log;

9) Переключившись на первое окно терминального мультиплексора, с помощью команды ssh -l student domen.name попытаться установить шифрованное соединени с удаленным сервером domen.name. Проследить передачу и прием пакетов между узлами в окне сетевого анализатора. Отметить взаимодействующие TCP-порты;

10) Подключившись к удаленной системе ввести пароль Password и выполнить команду uname -a, выведя информацию об удаленной системе;

11) Создать текстовый файл с содержанием ФИО и номера лабораторной работы на локальном узле и с помощью команды scp -v -o User=student/home/student/имя_файла domen.name:/home/student/

передать его по шифрованному каналу на удаленную систему. Проверить наличие копии переданного файла на удаленном узле, воспользовавшись файловым менеджером «Midnight Commander» (команда mc на удаленной системе);

12) Отключившись от удаленного узла (команда exit), на локальном хосте, сформировать зашифрованные ключи, воспользовавшись командой ssh-keygen;

13) Используя команду scp с указанием места расположения файла (публичного ключа) на локальной системе (/home/student/.ssh/key.pub), произвести его передачу по шифрованному туннелю на удаленный узел в заданный каталог /home/student/.ssh/ под именем authorized_keys. Проследить процесс пересылки пакетов между удаленными узлами в окне анализатора пакетов;

14) Воспользовавшись командой ssh -l student domen.name, снова сделать попытку подключения к удаленной системе. Отметить отличия в процедурах подключения и регистрации пользователя на удаленной системе;

15) Аналогично, с помощью команды scp, произвести повторную передачу текстового файла на удаленный узел. Убедиться в наличии переданной копии файла на удаленном хосте. Отметить отличия в процедуре передачи файла;

16) Остановить анализатор сетевых пакетов, воспользовавшись комбинацией Ctrl-c. Просмотреть содержимое файла ssh.log, отметить пакеты инициации сетевого взаимодействия и разрыва соединений TCP.

Выполнение практического задания.

1. SSH использует ассиметричное шифрование, суть которого заключается в наличии двух ключей: закрытого и открытого. Сгенерируем пару ключей, используя программу `ssh-keygen` (рисунок 1).

```
artem@ubuntuserver:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/artem/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match.  Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/artem/.ssh/id_rsa
Your public key has been saved in /home/artem/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:YB7+PcSKCxMvZnYYj4Vw4KDBhpacdX1OmRD3R9Aeq0I artem@ubuntuserver
The key's randomart image is:
+---[RSA 3072]----+
|* =. ++.o.o.     |
|o@ ....=. .o     |
|+ o . B  ...o    |
|   o = E ..o     |
|    + = S +      |
|     O + =       |
|    X = + o      |
|   + = .   .     |
|      .          |
+----[SHA256]-----+
artem@ubuntuserver:~$
```

Рисунок 1.

В результате создано два файла: id_rsa и id_rsa.pub. Открытый ключ хранится в файле /home/artem/.ssh/id_rsa закрытый — /home/artem/.ssh/id_rsa (рисунок 2).

```
artem@ubuntuserver:~$ ls -a
.   .bash_history  .bashrc  .config  .profile  .sudo_as_admin_successful  demo
..  .bash_logout   .cache   .local   .ssh      .viminfo                   dump.sql
artem@ubuntuserver:~$ cd .ssh
artem@ubuntuserver:~/.ssh$ ls -a
.  ..  id_rsa  id_rsa.pub  known_hosts
artem@ubuntuserver:~/.ssh$
```

Рисунок 2.

Скопируем содержимое файла id_rsa.pub на удалённую машину в файл /.ssh/authorized_keys (рисунок 3).

```
artem@ubuntuserver:~$ ssh-copy-id stud8@178.234.29.197
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/artem/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
stud8@178.234.29.197's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'stud8@178.234.29.197'"
and check to make sure that only the key(s) you wanted were added.

artem@ubuntuserver:~$
```

Рисунок 3.

Теперь выполним подключение с помощью клиента SSH (рисунок 4).

```
artem@ubuntuserver:~$ ssh 'stud8@178.234.29.197'
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-193-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

     https://microk8s.io/high-availability

15 packages can be updated.
0 updates are security updates.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** Требуется перезагрузка системы ***
Last login: Sun Jan 24 14:04:06 2021 from 217.107.199.150
$
```

Рисунок 4.

Передадим файл text.txt по шифрованному туннелю, воспользовавшись аутентфикацией по публичному ключу (рисунок 5).

```
artem@ubuntuserver:~$ scp ./text.txt stud8@178.234.29.197:
text.txt                                      100%   89     1.9KB/s   00:00
artem@ubuntuserver:~$
```

Рисунок 5.

2. Теперь, подключившись к удалённому серверу, используем полноэкранный консольный оконный менеджер screen.

3. Создадим новое окно и запустим анализатор трафика tcpdump, предварительно запустив терминальный мультиплексор tmux, с фильтром пакетов, получаемых и передаваемых от узла domen.name с TCP-портом источника и назначения 23. С помощью команды tee, выводим отфильтрованные IP-пакеты на терминал и сохраняем данные в файл telnet.log, в домашнем каталоге пользователя. Для этого следует воспользоваться командой `sudo tcpdump -l -v -nn tcp and src port 23 or dst port 23 | tee telnet.log;`

4-7. telnet – утилита, предназначеная для создания интерактивного соединения между удаленными компьютерами. Она работает по протоколу TELNET, но этот протокол поддерживается многими сервисами, поэтому её можно использовать для управления ими. Протокол работает на основе TCP, и позволяет передавать обычные строковые команды на другое устройство.

Попытка установить связь с удалённым сервером по протоколу TELNET оказалась неудачной, так как отсутствует связь с сервером по 23 порту. Об этом говорит ошибка, полученная в результате долгого ожидания ответа сервера (рисунок 6).

Рисунок 6.

8. Снова запустим анализатор сетевого трафика с фильтром пакетов, получаемых и передаваемых узлу domen.name с TCP-портом источника и назначения 22. Выводим отфильтрованные IP-пакеты на терминал и сохраняем данные в файл ssh.log. Для этого следует воспользоваться командой `sudo tcpdump -l -v -nn tcp and src port 22 or dst port 22 | tee ssh.log`;

9. Переключившись на первое окно с запущенным терминальным мультиплексором, попытаемся установить шифрованное соединение с удалённым сервером (рисунок 7).

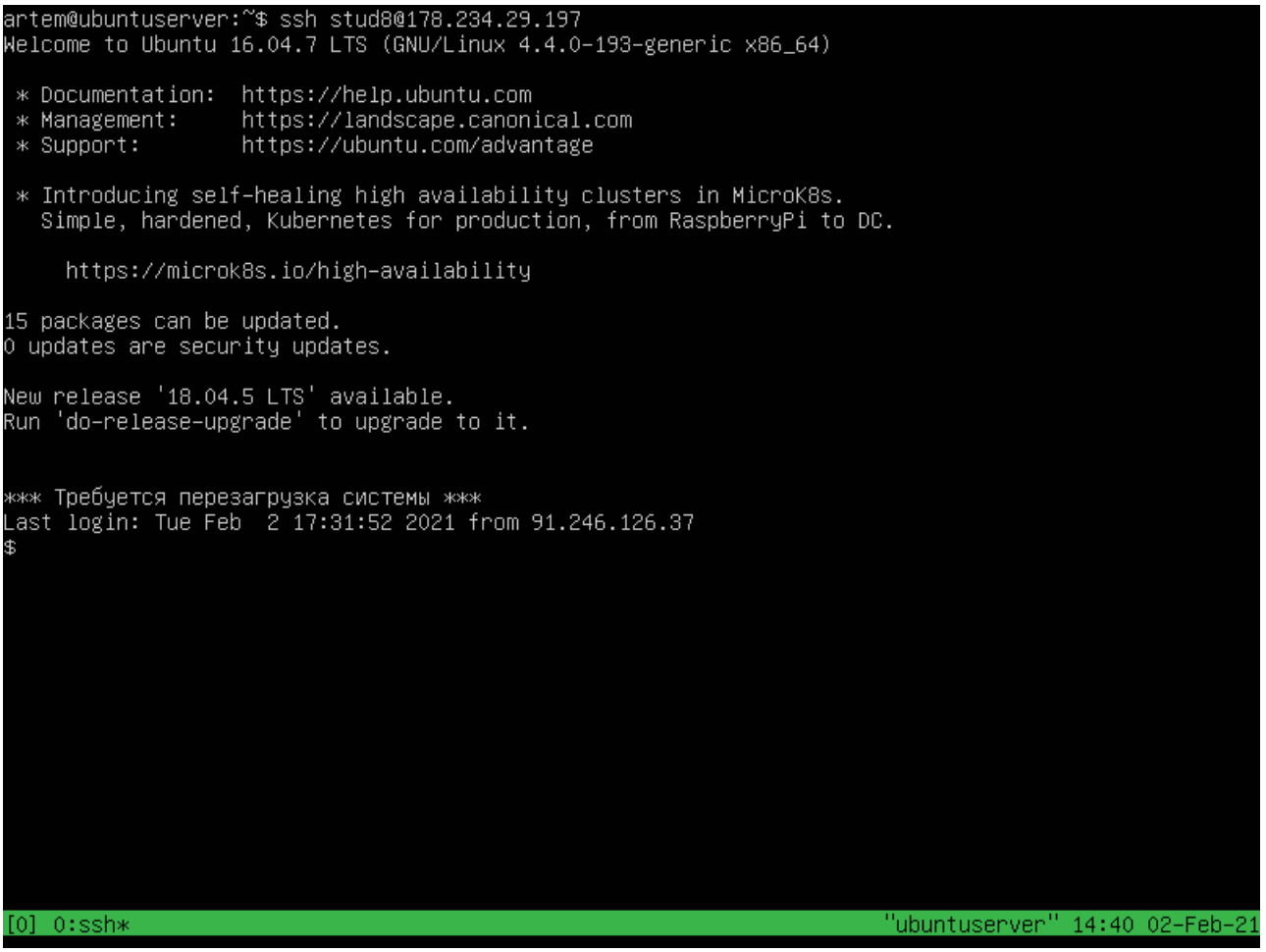

Рисунок 7.

Вернувшись к окну с запущенным анализатором сетевого трафика, заметим, что в процессе соединения с удалённым сервером была осуществлена передача 71-ого IP-пакета между сервером по 22-ому порту и клиентом по 39468-ому (рисунок 8).

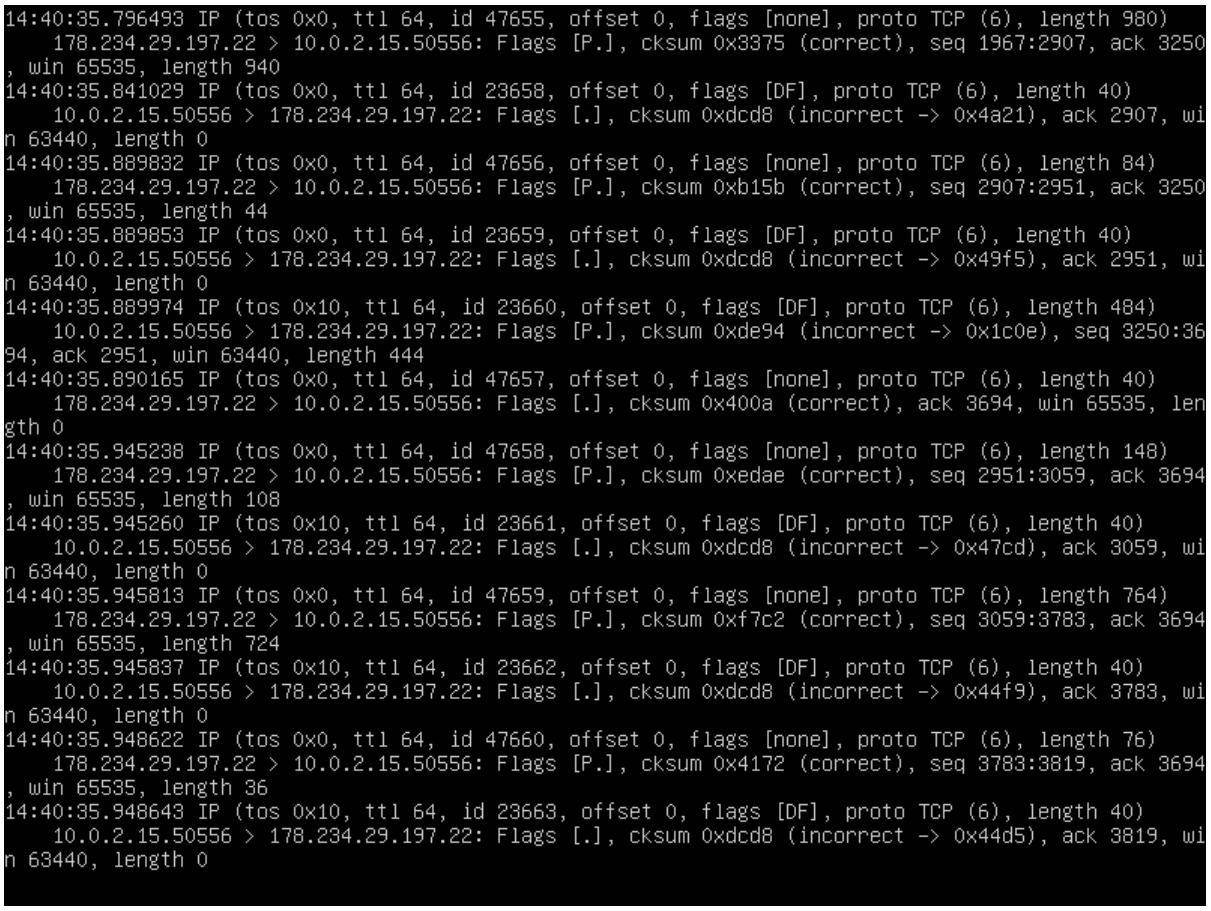

Рисунок 8.

10. После подключения к удаленной системе и ввода пароля выводим полную информацию об удалённой системе (рисунок 9).

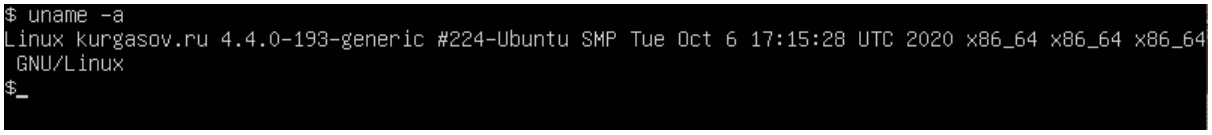

Рисунок 9.

11. Теперь создадим на локальном узле текстовый файл lr7.txt, содержащий ФИО и номер лабораторной работы, и с помощью команды `scp -v -o User=stud8 ./lr7.txt 178.234.29.197:` передадим его по шифрованному каналу на удалённую систему.

Проверим наличие копии переданного файла на удаленном узле, воспользовавшись файловым менеджером "Midnight Commander"(рисунок 10)

Рисунок 10.

12. Отключившись от удалённого узла, вновь сгенерируем зашифрованные ssh-ключи key_rsa и поместим их в `/home/artem/.ssh`.

13. Используя команду scp с указанием места расположения файла key_rsa.pub на локальной системе `./.ssh/key_rsa.pub`, произведём его передачу по шифрованному туннелю на удалённый узел в заданный каталог `/home/stud8/.ssh/authorized_keys`.

14. Вновь подключаемся к удалённой системе. Заметем, что отличий в процедурах подключения нет и, хотя в процедуре регистрации и используются разные команды, в целом они мало отличаются и заключаются в том, чтобы передать публичный ключ на сервер.

15. Аналогично, с помощью команды scp произведём повторную передачу текстового файла lr7.txt на удалённый узел.

Файл был успешно передан, а различий в передаче обнаружено небыло.

16. Остановим анализатор сетевых пакетов и просмотрим содержимое файла ssh.log. Утилита tcpdump позволяет проверять заголовки пакетов TCP/IP и выводить одну строку для каждого из пакетов. Флаги TCP указывают на состояние соединения и могут содержать более одного значения. Рассмотрим на следующем примере: (рисунок 19). Флаг [P.] устанавливается при передаче пользовательских данных между удалёнными узлами(клиентом и сервером), флаг [F.] - устанавливается при нормальном закрытии соединения.

Вопросы для самопроверки.

Что такое ключ SSH? В чём преимущество их использования?

SSH сервер может выполнять аутентификацию пользователей с помощью различных алгоритмов. Хоть самым популярным и является аутентификация по паролю, так как она достаточно проста, но более безопасным и надёжным является аутентификация по ключу SSH.

SSH-ключи используются для идентификации клиента при подключении к серверу по SSH-протоколу. SSH-ключи представляют собой пару — закрытый и открытый ключ. Закрытый должен храниться в закрытом доступе у клиента, открытый отправляется на сервер и размещается в файле `authorized_keys`.

Как сгенерировать ключи ssh в разных ОС?

Генерация ключей ssh в linux.

Пару ключей для использования ssh можно сгенерировать с помощью команды `ssh-keygen`. Если не задавать других параметров, ключи сохранятся по пути `/.ssh/` в формате <name> для секретного ключа и <name>.pub для открытого. Ключи также можно защитить паролем при создании. Если указать пароль пустым, он не будет использоваться. Сменить используемый пароль можно с помощью `ssh-keygen -p`.

Генерация ключей ssh в Windows.

В ОС Windows подключение к удалённым серверам по SSH возможно с помощью клиента Putty. Для генерации ssh-ключа необходимо запустить файл puttygen.exe. Выбрать тип ключа SSH-2 RSA и длину 2048 бит, а затем сгенирировать (кнопка Generate) и сохранить пару ключей на локальной машине (кнопки Save public key и Save private key).

Возможно ли из "секретного"ключа сгенерировать "публичный"и/или наоборот?

Из секретного ключа генерируется публичный, но наоборот сделать это невозможно.

Будут ли отличаться пары ключей, сгенерированные на одном ПК несколько раз с исходными условиями (наличие/отсутствие пароля на "секретный"ключ и т.п.)

Генерируемые ssh-ключи являются уникальными и не могут совпадать при их повторном создании. Дополнительная возможность указания пароля на "секретный"ключ является мерой для большей устойчивости ко взлому и на уникальность ключей не влияет.

Перечислите доступные ключи для ssh-keygen.exe

Программа ssh-keygen можен генерировать четыре типа ключей:

1) dsa;
2) ecdsa;
3) ed25519;
4) rsa;

Чтобы выбрать любой из этих типов, используется опция -t. Тип rsa подразумевается по умолчанию (то есть генерацию ключей можно запустить без опции -t).

Можно ли использовать один "секретный"ключ доступа с разных ОС, установленных на одном ПК/на разных ПК?

Возможно.

Возможно ли организовать подключение "по ключу"ssh к системе с ОС Windows, в которой запущен OpenSSH сервер?

Возможно.

Какие известные Вам сервисы сети Интернет позволяют организовать доступ к ресурсам посредством SSH ключей?

WWW-серверы, FTP-серверы, почтовики, шлюзы

Определите основные цели и задачи решаемые с помощью ПО удаленного доступа?

С помощью ssh можно производить удаленное управление хостом и туннелирование TCP-соединений. В отличие от telnet и rlogin весь трафик, передаваемый по ssh, шифруется. SSH позволяет безопасно передавать в незащищённой среде практически любой другой сетевой протокол. Таким образом, можно не только удалённо работать на компьютере через командную оболочку, но и передавать по шифрованному каналу звуковой поток или видео.

Выделите отличительные особенности между режимами работы удаленного доступа по протоколам TELNET и SSH?

Для удаленного доступа к Linux используются два протокола **telnet** и **SSH**.

Telnet - протокол линии передачи данных Интернет, который даёт возможность компьютеру функционировать как терминал, работающий под управлением удалённого компьютера.

Нежелательно использование протокола telnet в системах, для которых важна безопасность, таких как общественный Интернет. Сеансы telnet не поддерживают шифрование данных. Это означает это любой, кто имеет доступ к любому маршрутизатору, коммутатору или шлюзу в сети между двумя удалёнными компьютерами, соединенными сеансом связи по протоколу telnet, может перехватить проходящие пакеты и легко получить логин и пароль для доступа в систему (или завладеть любой другой информацией, которой обмениваются эти компьютеры).

SSH - (Secure Shell) — сетевой протокол, позволяющий производить удалённое управление компьютером и передачу файлов. Сходен по функциональности с протоколом telnet , однако использует алгоритмы шифрования передаваемой информации.

Недостатки telnet привели к очень быстрому отказу от использования этого протокола в пользу более безопасного и функционального протокола SSH. SSH предоставляет все те функциональные возможности, которые представлялись в telnet, с добавлением эффектного кодирования с целью предотвращения перехвата таких данных, как логины и пароли. Введенная в протоколе SSH система аутентификации с использованием публичного ключа гарантирует, что удаленный компьютер действительно является тем, за кого себя выдает.

Опишите способы установления соединения при использовании протокола SSH? Охарактеризуйте положительные и отрицательные аспекты приведенных методов.

Соединение по логину-паролю, и соединение по ключу.

- Логин-пароль:
  плюсы - для авторизации необходимо знать пару логин пароль, можно авторизоваться с любой системы;
  минусы - данный способ уязвим для взлома(есть возможность подобрать логин и пароль).
- Ключ:
  плюсы - данный способ устойчив к взломам и гарантирует высокую надёжность;
  минусы - на устройстве необходимо наличие публичного ключа.

Перечислите распространенные сетевые службы, основанные на использовании шифрованного соединения по протоколу SSH? Приведите пример использования службы передачи файлов по безопасному туннелю?

Область применения протокола SSH практически неограничена. Исходя из его основной функции - удаленного входа в операционную систему, протокол используют:

1) системные администраторы для удаленной настройки компьютеров локальной сети;
2) для настройки почтовых служб (повышает безопасность данных);
3) для скрытого обмена внутри сети массивными файлами;
4) для интернет-игр;

Вывод

В ходе лабораторной работы было изучено программное обеспечение удалённого доступа к распределённым системам обработки данных.

Список литературы

[1] Львовский, С.М. Набор и верстка в системе LaTeX [Текст] / С.М. Львовский. М.: МЦНМО, 2006. — 448 с.

[2] Хабр. Используем tcpdump для анализа и перехвата сетевого трафика: https://habr.com/ru/company/alexhost/blog/531170/ (дата обращения: 30.01.2021). - Текст: электронный.

[3] Losst. Удалённое управление telnet: https://losst.ru/kak-polzovatsya-telnet6_Удалённое_управление_telnet (дата обращения: 30.01.2021). - Текст: электронный.