Лабораторная работа

Работа с SSH

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

Описание: Программное обеспечение систем обработки удаленного доступа строится на основе архитектуры «клиент-сервер». После успешной авторизации клиентской части ПО пользователя, серверный процесс передает управление указанной командной оболочки операционной системы. Окружение пользователя в случае доступа к удаленной системе, практически полностью аналогично работе в окружении локальной системы. Наиболее распространенными реализациями программного обеспечения обработки удаленного доступа являются службы «telnet» и «ssh». В работе каждой из указанных служб существует ряд определенных особенностей. Протокол прикладного уровня эталонной сетевой модели OSI «TELNET» реализует сетевой текстовый (псевдографический) интерфейс между удаленными хостами. В качестве конечных точек взаимодействия могут выступать виртуальные терминалы, системные процессы и пр. Серверный процесс службы telnet по-умолчанию ожидает соединений на TCP-порту 23. Проведя регистрацию пользователя (логин – пароль), служба предоставляет доступ к интерфейсу командной строки. Следует отметить, что протокол TELNET не имеет встроенной поддержки шифрования и является **критически уязвимым** к проведению любых видов сетевых атак. Применение и исследование службы telnet оправдано с исторической и методологической точки зрения. Кроме указанных причин, клиент telnet, также применяется для исследования доступных сетевых служб на удаленных хостах. Реализация протокола TELNET, чаще осуществляется серверной службой telnetd и клиентом telnet.

В отличии от службы telnet, протокол SSH предоставляет шифрованное транспортное соединение к удаленной системе обработки данных. Передаваемая информация подвергается симметричному блочному шифрованию, с использованием одного из стойких алгоритмов шифрования — Blowfish или Triple DES (3DES). Особенностью работы протокола SSH является создание частного и публичного зашифрованных ключей. В процессе установления соединения и проверки подключения, удаленные узлы обмениваются публичными ключами, подтверждая т. о. свою подлинность. Серверный процесс ожидает подключений на TCP-порту 22. После процедур обмена шифрованными ключами и регистрации пользователя, сервер передает управление интерфейсу командной оболочки. Любая информация передаваемая по протоколу SSH инкапсулируется в транспортные сегменты, подвергается стойкому шифрованию и с помощью процедур сетевого уровня передается на удаленную систему.

Существует несколько типов подключения к удаленной системе с использованием службы ssh. В общем случае, для создания сетевого подключения к удаленной системе обработки данных предназначена команда ssh, в качестве одного из аргументов которой служит IP-адрес или доменное имя удаленной системы

user@ubuntu [~]: ssh 182.161.150.92

В примере происходит попытка установления соединения с узлом по IP адресу, где в качестве имени пользователя будет использоваться логин user. В случае успешной попытки подключения к удаленному узлу, необходимо указать пароль. Опция -l, команды ssh, используется в том случае, если необходимо произвести подключение к удаленной системе под другим логином и паролем: **user@host [~]: ssh -l admin 182.161.150.92**

Краткий список опций команды SSH	
Опция	Назначение
-1	Использовать протокол SSH версии 1

-2	Использовать протокол SSH версии 2
-4	Использовать только адреса IPv4
-6	Использовать только адреса IPv6
-c blowfish|3des|des	Указать алгоритм для шифрования сеанса
-C	Использовать сжатие данных
-l пользователь	Войти на удаленную систему под учетной записью заданного пользователя
-p	Задать порт для подключения на удаленной системе
-q	Не выводить подробную информацию о подключении
-v	Выводить подробную информацию о подключении
-V	Вывести версию ПО

Приведенные выше примеры описывают аутентификацию с ключом узла. Это означает, что соединение устанавливается на основании публичного ключа удаленного узла и регистрационной информации пользователя. Однако для создания более надежного подключения применяется метод аутентификации с публичным ключом. В этом случае необходимо сгенерировать собственную пару из публичного и частного ключа, а затем распространить публичный ключ на удаленные узлы. Т. о. становится возможным проводить защищенную аутентификацию без использования регистрационного пароля пользователя системы удаленной обработки данных.

Для выполнения данных процедур необходимо использовать команду ssh-keygen, которая сформирует пару ключей в корневом каталоге пользователя. Данные хранящиеся в файле key.pub следует передать на удаленный узел и записать (дописать) в файл /.ssh/authorized_keys. /key - имя, данное при генерации/

user@host [~]: ssh_keygen

Generating public / private rsa key pair.

Далее необходимо передать публичный ключ локальной системы на удаленный узел. С помощью команды scp можно произвести передачу файлов по шифрованному транспортному соединению ssh. Команда scp поддерживает множество способов передачи данных, однако в данном случае, в качестве аргументов достаточно указать систему отправки и удаленный узел приемки файла:

user@host [~]: scp~/.ssh/key.pub remote :~/.ssh/authorized_keys

В качестве первого аргумента команды scp выступает файл key.pub, который необходимо передать узлу получателю remote, в каталог .ssh под именем authorized_keys. По завершению указанных процедур, становится возможным подключится к удаленной системе, произведя авторизацию по отпечатку публичного ключа.

Опция	Назначение
-4	Использовать только адреса IPv4
-6	Использовать только адреса IPv6
-B	Пакетный режим передачи
-C	Использовать сжатие данных
-P	Подключаться к указанному порту на удаленной системе
-q	Не показывать ход выполнения передачи данных
-r	Выполнять рекурсивное копирование каталогов
-v	Режим подробного вывода

Краткий список опций команды scp

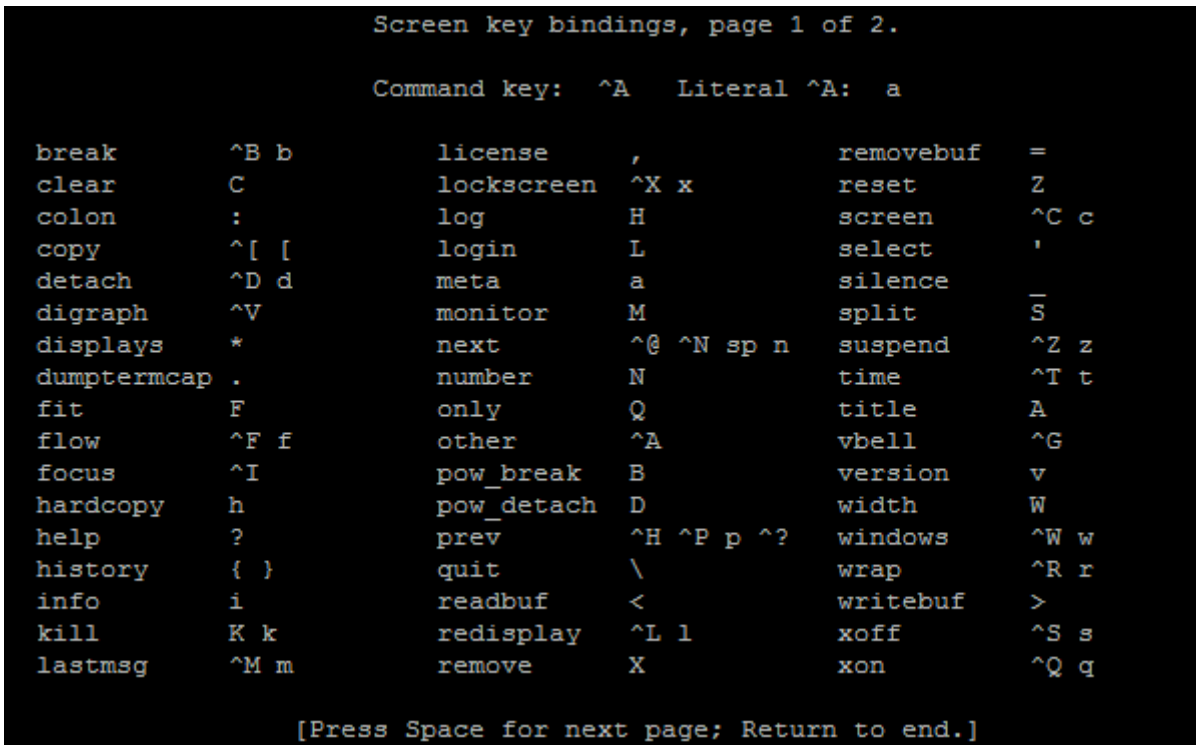
Беспарольное подключение по отпечатку публичного ключа:
user@host [~]: ssh remote

Рекомендации по безопасности использования SSH:

1. Запрещение удаленного root-доступа.
2. Запрещение подключения с пустым паролем или отключение входа по паролю.
3. Выбор нестандартного порта для SSH-сервера.
4. Использование длинных SSH2 RSA-ключей (более 2048 бит).
5. Ограничение списка IP-адресов, с которых разрешен доступ. Например, настройкой файрвола.
6. Запрещение доступа с некоторых, потенциально опасных адресов.
7. Отказ от использования распространенных или широко известных системных логинов для доступа по SSH.
8. Регулярный просмотр сообщений об ошибках аутентификации.
9. Установка детекторов атак (IDS, Intrusion Detection System).
10. Использование ловушек, подделывающих SSH-сервис (honeypots)

SCREEN - полноэкранный консольный оконный менеджер с поддержкой скроллинга и поиска в окне и функцией копирования-вставки между ними.
Его плюсы:
+ Вы можете в любой момент отсоединиться от своего screen`а и закрыть сеанс работы в шеле. После этого Вы можете присоединившись к screen`у вновь продолжить свою работу с того места где Вы остановились.
Например, допустим вы начали пересобирать ядро, или запустили какой-то длительный процесс, но дождаться его завершения нет возможности. Screen позволяет отключиться от текущего сеанса не прерывая выполняемой работы. В любой момент вы имеете возможность продолжить работу так, как будто вы и не отключались. Кроме того, при внезапном разрыве ssh соединения с сервером screen-сессия не прерывается, что очень удобно при работе по слабым каналам связи (к примеру GSM модемы).
+ возможность управления несколькими консольными окнами.
Суть в том, что screen позволяет, используя единственную ssh сессию, создать несколько окон терминалов, с легкостью переключаться между ними и выполнять различные задачи в каждом из них параллельно.
+ одновременное подключение нескольких пользователей к одной активной сессии screen.
Где такая функция может понадобиться? Например в таком режиме работал консольный icq клиент службы техподдержки, где любой консультант мог отвечать на вопросы, подключившись к активной сессии screen. Так же его вполне можно использовать для обучения новых сотрудников.

1. Использование
   После авторизации в системе необходимо набрать screen
   На первый взгляд ничего не изменилось, однако, программа уже запущена. Убедиться в этом позволяет Ctrl+A, затем ?



Все управляющие команды начинаются с Ctrl+a

Конфигурационный файл **.screenrc** находиться в домашнем каталоге текущего пользователя. В случае его отсутствия, можно скопировать файл общесистемный screenrc который находиться в каталоге /etc.

Все опции можно изменить во время работы. Для этого нажмите Ctrl+a : и введите название параметра и его значение.

Разберем некоторые директивы:

- vbell off - управляет визуальным звонком. Если данный параметр будет включен (on) то звонок будет отображаться как вспышка на экране.
- activity 'activity in window %n' - сообщение которое будет выводиться при включенном режиме мониторинга за окном. Полезно если Вы ждете какого либо действия в окне.
- bell_msg 'bell in window %n' - сообщение которое выведется на Ваш экран в случае получения screen`ом звукового сигнала в каком либо окне.
- nethack on - изменяет стиль текста выводимых сообщений на стиль знаменитой игрушки NetHack. Почувствуйте себя в подземельях... ;)
- autodetach on - если по какой то причине соединение с управляющим процессом будет потеряно, то после востановления работа в screen может быть возобновлена. В обратном случае (off) - screen будет уничтожен со всеми дочерними окнами и процессами.
- startup_message off - выключает сообщение об авторских правах при первом запуске screen`a.
- defscrollback 10000 - количество строк по умолчанию для буфера прокрутки.
- caption always - показывает заголовки окна в строке статуса.
- caption string "%{rk} %c %{dd} %{+b M}%n %{-b dd}%-w%{+b B.}%n* %t%{-}%+w%<" - форматирование строки статуса. Данный набор символов приведет к тому что в строке статуса будет отображаться время и цветом выделяться активное окно.

После запуска screen создаст одно окно с Вашим шелом. В последствии вы сможете создать дополнительные окна. Все нажатия клавиш передаются текущей программе в окне. Ограничение накладывается только на управляющую последовательность самого менеджера. Данная последовательность Ctrl+a. Для того что бы передать приложению данную последовательность Вам нужно нажать Ctrl+a и сразу a. Тип терминала должен быть VT100 совместим для правильной передачи нажатий при удаленной работе.

Тип терминала передаваемый приложению в окне screen - так и называться screen. Если Ваше приложение не поддерживает данный тип - его всегда можно изменить путем изменения переменной TERM.

КРАТКАЯ СВОДКА КОМБИНАЦИЙ КЛАВИШ ПРИ РАБОТЕ

Для создания нового окна - Ctrl+a c (create).

Для переключения между окнами - Ctrl+a a - между последним активным.

Ctrl+a <НОМЕР> - выбор окна по номеру. Ctrl+a (p|n) - циклическое перемещение между окнами. p - prev, n - next. Ctrl+a " - список окон для переключения.

Управление окнами - Ctrl+a A - изменить заголовок окна. Аналогично вводу команды title при нажатии Ctrl+a :

- Ctrl+a C - очистить окно.
- Ctrl+a F - подогнать размер окна под текущий размер терминала.
- Ctrl+a H - протоколирование окна в файл screenlog.<НОМЕР ОКНА>
- Ctrl+a K - уничтожить окно.
- Ctrl+a M - режим слежения за активностью в окне. Если в момент этого вы находитесь в другом окне - в подсказке будет выведено:activity in window <НОМЕР ОКНА>
- Ctrl+a r - переключение режима переноса по словам. (wrap)
- Ctrl+a S - очень интересный режим работы. Сплит. То-есть текущее окно разделяется на две части и в обоих можно открыть по новому окну.

Переключение между окнами Ctrl+a; TAB, выход из режима сплит - Ctrl+a Q.

Общие команды

- Ctrl+a ? - помощь
- Ctrl+a Esc - режим скроллинга. Он же режим копирования. Для копирования подведите курсор к нужному месту и нажмите пробел.
- Ctrl+a] - Вставка выделенной области.
- Ctrl+a x - Запереть менеджер. При вкомпиленной поддержке PAM - для разблокировки нужно ввести пароль пользователя от которого запущен менеджер. В обратном случае пароль для разблокировки будет запрошен при блокировании.

НАИБОЛЕЕ ЧАСТО ПРИМЕНЯЕМЫЕ ОПЦИИ КОМАНДНОЙ СТРОКИ.

- rd - подключиться к screen. Сделать deatach для остальных сессий.
- list/-ls - список запущенных менеджеров.
- dm - запуск screen в режиме deatach. Полезно для init скриптов или скриптов вообще.
- wipe - удалить сведения о запущенных менеджерах. Полезно в случае потери менеджера, но сохранения информации о нем.
- x - присоединиться к screen. Присоединение осуществляется даже в случае существующих соединений. Полезно при работе с одним screen из разных окружений. Например, один screen и на X и на консоль. ;)

Задание к Лабораторной работе

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.

1. 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;**

2. 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/
3. Воспользовавшись окном сетевого монитора, анализировать прохождение сетевых пакетов между узлами назначения. Отметить пакеты инициации соединения telnet;
4. 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;
5. 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;
6. 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;**
7. 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;
8. Подключившись к удаленной системе ввести пароль Password и выполнить команду uname -a, выведя информацию об удаленной системе;
9. 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);
10. Отключившись от удаленного узла (команда exit), на локальном хосте, сформировать зашифрованные ключи, воспользовавшись командой ssh-keygen;
11. 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;

12. 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;
13. 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;
14. Остановить анализатор сетевых пакетов, воспользовавшись комбинацией Ctrl-c. Просмотреть содержимое файла ssh.log, отметить пакеты инициации сетевого взаимодействия и разрыва соединений TCP.

Отчет: Сформировать отчет в электронном виде. К отчету приложить именные текстовые файлы telnet.log и ssh.log.

Контрольные вопросы:

1) Определите основные цели и задачи решаемые с помощью ПО удаленного доступа?
2) Выделите отличительные особенности между режимами работы удаленного доступа по протоколам TELNET и SSH?
3) Опишите способы установления соединения при использовании протокола SSH? Охарактеризуйте положительные и отрицательные аспекты приведенных методов.
4) Основываясь на заданиях лабораторной работы, приведите практический пример использования систем удаленного доступа?
5) Перечислите распространенные сетевые службы, основанные на использовании шифрованного соединения по протоколу SSH? Приведите пример использования службы передачи файлов по безопасному туннелю?

Материалы по теме:

SSH keys