

УВАЖАЕМЫЕ СТУДЕНТЫ!

ВАМ НЕОБХОДИМО ВЫПОЛНИТЬ СЛЕДУЮЩЕЕ:

1. Ознакомиться с теорией, составить и ответить на вопросы.
2. Предоставит отчет конспекта лекции прислать в виде скриншото в течении трех дней .
3. Отправить преподавателю на почту v.vika2014@mail.ru и указать свою Ф.И.О, группу, и название дисциплины тел 072-17-44-9-22

Тема: Алгоритм технологии установки и настройки FTP-сервера и Web-сервера

Настройка ftp-сервера

FTP (англ. File Transfer Protocol - протокол передачи файлов) - протокол, предназначенный для передачи файлов в сетях передачи данных. FTP позволяет подключаться к серверам FTP, просматривать содержимое каталогов и загружать файлы с сервера или на сервер; кроме того, возможен режим передачи файлов между серверами.

FTP является одним из старейших прикладных протоколов, появившимся задолго до HTTP, в 1971 году. Он и сегодня широко используется для распространения программного обеспечения и доступа к удалённым хостам.

Протокол FTP относится к протоколам прикладного уровня и для передачи данных использует транспортный протокол TCP. Команды и данные, в отличие от большинства других протоколов, передаются по разным портам. Исходящий порт 20, открываемый на стороне сервера, используется для передачи данных, порт 21 для передачи команд. Порт для приема данных клиентом определяется в диалоге согласования. В случае, если передача файла была прервана по каким-либо причинам, протокол предусматривает средства для докачки файла, что бывает очень удобно при передаче больших файлов.

Vsftpd (Very Secure FTP Daemon или Очень Защищенный FTP Демон) является одним из самых простых в конфигурировании и наиболее часто используемым FTP сервером. Vsftpd обслуживает ftp серверы debian, redhat, ubuntu и прочих крупных компаний. Благодаря предельной простоте настройки, поднятие ftp сервера с помощью vsftpd редко занимает более 5 - 10 минут.

В данной лабораторной работе предполагается показать принцип создания файлового сервера, на который все пользователи смогут складывать файлы, удалять их, создавать директории и т.д.

Установка vsftpd

Установка vsftpd приведена на рис. 1. Перед установкой необходимо проверить, что есть соединение с Internet.

```
work@work:~$ sudo apt-get install vsftpd
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
```

Рис. 1. Установка ftp

Настройка vsftpd

Конфигурирование vsftpd осуществляется редактированием файла /etc/vsftpd.conf (Рис. 4.2). Комментариев (при минимальном знании английского) обычно достаточно, чтобы разобраться что к чему:

- anon_root - директория для анонимных пользователей (/var/ftp/ по умолчанию в большинстве дистрибутивов);
- anonymous_enable - разрешить доступ анонимным пользователям;
- local_enable - разрешить доступ локальным пользователям;
- write_enable - разрешить запись;
- anon_upload_enable - разрешить запись анонимным пользователям

Таким образом, можно отредактировать эти записи в конфиге следующим образом (не стоит удалять остальные опции, если вы не знаете, что они делают):

```
#возможность работы в автономном режиме
listen=YES
```

#позволяем анонимных пользователей, учетки anonymous и ftp являются синонимами

anonymous_enable=YES

#разрешаем локальных пользователей (локальные пользователи - это те, которые

#зарегистрированы в системе, то есть на них есть учетные записи)

local_enable=YES

#разрешаем любые формы записи на FTP сервер

write_enable=YES

#разрешаем анонимным пользователям upload

anon_upload_enable=YES

#разрешаем анонимным пользователям создавать директории

anon_mkdir_write_enable=YES

#разрешаем анонимным пользователям переименовывать файлы

anon_other_write_enable=YES

#у анонимов пароль спрашивать не будем

no_anon_password=YES

#директория для доступа анонимных пользователей (если пользователь присутствует)

anon_root=/home/ftp/

#разрешаем соединение по 20 порту

connect_from_port_20=YES

#поддержка древних FTP клиентов

async_abor_enable=YES

#используем родное время, а не GMT

use_localtime=YES

#небольшое приветствие

ftpd_banner=Hello! We come in peace!

#возможность работы как фоновый процесс

background=YES

Должны ли пользователи находиться только в своих директориях
YES/NO chroot_local_user=YES

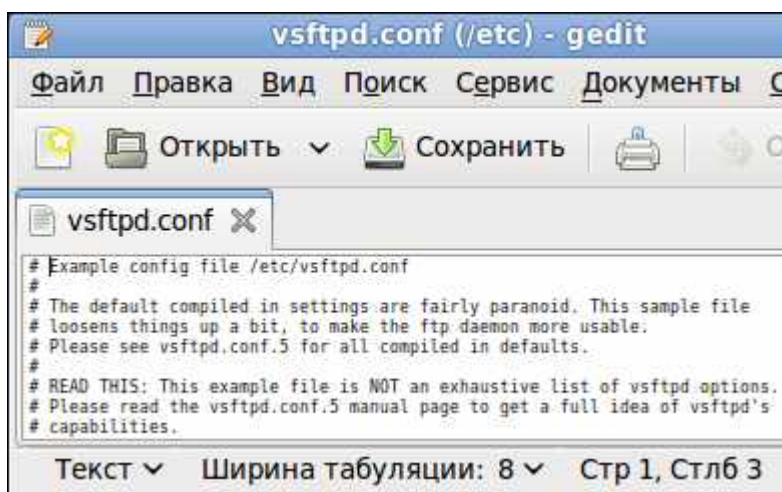


Рис.2. Файл конфигурации ftp

Telnet

TELNET (англ. TErminaL NETwork) - сетевой протокол для реализации текстового интерфейса по сети (в современной форме - при помощи транспорта TCP). Название «telnet» имеют также некоторые утилиты, реализующие клиентскую часть протокола. Современный стандарт протокола описан в RFC 854.

Выполняет функции протокола прикладного уровня модели OSI.

Назначение протокола TELNET в предоставлении достаточно общего, двунаправленного, восьмибитного байт-ориентированного средства связи. Его основная задача заключается в том, чтобы позволить терминальным устройствам и терминальным процессам взаимодействовать друг с другом. Предполагается, что этот протокол может быть использован для связи вида терминал-терминал («связывание») или для связи процесс-процесс («распределенные вычисления»).

В протоколе не предусмотрено использование ни шифрования, ни проверки подлинности данных. Поэтому он уязвим для любого вида атак, к которым уязвим его транспорт, то есть протокол TCP. Для функциональности удалённого доступа к системе в настоящее время применяется сетевой протокол SSH (особенно его версия 2), при создании

которого упор делался именно на вопросы безопасности. Так что следует иметь в виду, что сессия Telnet весьма незащищена, если только не осуществляется 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.

Сетевой протокол ssh

SSH - это специальный сетевой протокол, позволяющий получать удаленный доступ к компьютеру с большой степенью безопасности соединения.

В основном, ssh реализован в виде двух приложений – ssh-сервера и ssh-клиента. В Ubuntu используется свободная реализация клиента и сервера ssh - OpenSSH. При подключении клиент проходит процедуру авторизации у сервера и между ними устанавливается зашифрованное соединение. OpenSSH сервер может работать как с протоколом ssh1, так и с протоколом ssh2. В настоящее время протокол ssh1 считается небезопасным, поэтому его использование крайне не рекомендуется.

Установить OpenSSH можно так:

```
work@work:~$ sudo aptitude install ssh
```

Рис. 3. Установка OpenSSH

Метапакет ssh содержит в себе и клиент и сервер, при этом скорее всего будет установлен только сервер, т. к. клиент часто бывает установлен в Ubuntu по умолчанию.

SSH сервер автоматически прописывается в автозагрузку при установке. Управлять его запуском/остановкой или перезапуском можно при помощи команд:

```
sudo service ssh stop|start|restart
```

Основным файлом конфигурации ssh-сервера является файл /etc/ssh/sshd_config, который должен быть доступным для чтения/редактирования только суперпользователю. После каждого изменения

этого файла необходимо перезапустить ssh-сервер для применения изменений.

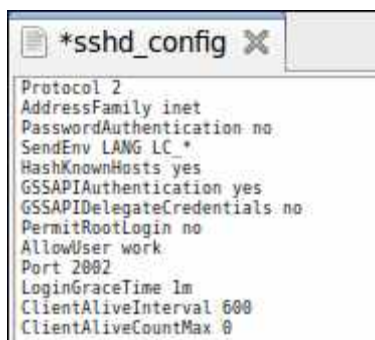
Сам по себе, неправильно настроенный ssh сервер - огромная уязвимость в безопасности системы, т. к. у возможного злоумышленника есть возможность получить практически неограниченный доступ к системе. Помимо этого, у sshd есть много дополнительных полезных опций, которые желательно включить для повышения удобства работы и безопасности.

Для правильной настройки ssh с точки зрения безопасности необходимо отредактировать всего семь параметров:

1. `PermitRootLogin` – отключение возможности авторизации под суперпользователем;
2. `AllowUsers`, `AllowGroups` - предоставление доступа только указанным пользователям или группам;
3. `DenyUsers`, `DenyGroups` - блокировка доступа определенным пользователям или группам;
4. `Port` - изменение порта SSHD;
5. `LoginGraceTime` - изменение времени ожидания авторизации;
6. `ListenAddress` - ограничение авторизации по интерфейсу;
7. `ClientAliveInterval` - рассоединение при отсутствии активности в шелле.

Сменить стандартный порт (22) на котором слушает sshd. Это связано с тем, что многочисленные сетевые сканеры постоянно пытаются соединиться с 22-м портом и как минимум получить доступ путем перебора логинов/паролей из своей базы. Даже если у вас и отключена парольная аутентификация - эти попытки сильно засоряют журналы и (в большом количестве) могут негативно повлиять на скорость работы ssh-сервера. Если же вы по какой либо причине не желаете изменить стандартный порт вы можете использовать как различные внешние утилиты для борьбы брутфорсерами, например `fail2ban`, так и встроенные, такие как `MaxStartups`.

По умолчанию root-доступ разрешен. Это означает, что клиент при подключении в качестве пользователя может указать root, и во многих случаях получить контроль над системой. При условии, что по умолчанию в Ubuntu пользователь, добавленный при установке системы имеет возможность решать все административные задачи через sudo, создавать возможность root доступа к системе как минимум странно. Рекомендуется отключить эту опцию совсем.

A screenshot of a terminal window showing the configuration for the sshd service. The window title is '*sshd_config'. The content is as follows:

```
Protocol 2
AddressFamily inet
PasswordAuthentication no
SendEnv LANG LC *
HashKnownHosts yes
GSSAPIAuthentication yes
GSSAPIDelegateCredentials no
PermitRootLogin no
AllowUser work
Port 2882
LoginGraceTime 1m
ClientAliveInterval 600
ClientAliveCountMax 0
```

Рис. 4. Файл конфигурации ssh

Разрешенная по умолчанию парольная аутентификация является практически самым примитивным способом авторизации в ssh. С одной стороны это упрощает конфигурацию и подключение новых пользователей (пользователю достаточно знать свой системный логин/пароль), с другой стороны пароль всегда можно подобрать, а пользователи часто пренебрегают созданием сложных и длинных паролей. Специальные боты постоянно сканируют доступные из интернета ssh сервера и пытаются авторизоваться на них путем перебора логинов/паролей из своей базы. Настоятельно не рекомендуется использовать парольную аутентификацию.

Как уже было сказано, ssh может работать с протоколами ssh1 и ssh2. При этом использование небезопасного ssh1 крайне не рекомендуется.

В конечном итоге файл конфигурации должен выглядеть так, как на рис. 4.

Для удаленного доступа с операционной системы Windows необходимо установить на ней специальный клиент – putty (рис 5).

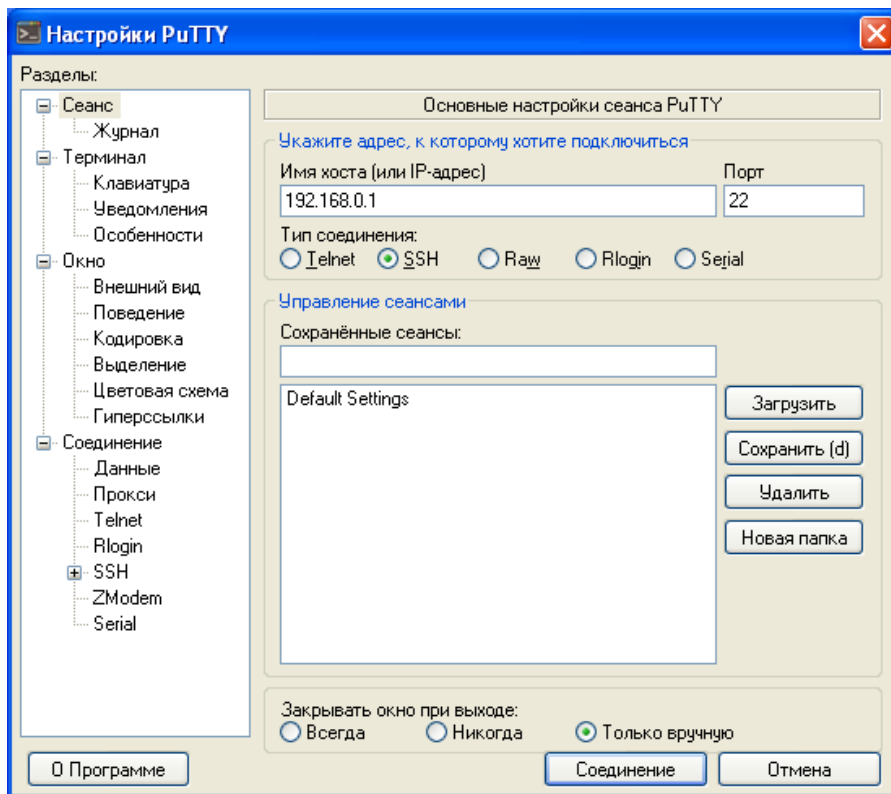


Рис. 5. PuTTY

Для настройки сессии введите IP хоста (192.168.0.1). Так же настройте кодировку в пункте Translation, поменяв её на UTF-8.

Веб-сервер

Apache HTTP-сервер – свободный веб-сервер. Apache является кроссплатформенным программным обеспечением, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.

Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает IPv6.

Для установки apache2 введите команду, представленную на рис. 6.

```
work@work:~$ sudo apt-get install apache2
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
```

Рис.6. Установка apache2

Файлы конфигурации Apache2 находятся в директории: `/etc/apache2`:

- `conf.d/`
- `sites-available/`
- `sites-enabled/`
- `mods-available/`
- `mods-enabled/`
- `apache2.conf`
- `envvars`
- `httpd.conf`
- `ports.conf`

В Ubuntu основным файлом настройки Apache2 является `apache2.conf`. Он играет роль системного файла, в котором собраны основные и самые важные настройки сервера.

Файл `httpd.conf` - пустой и предназначен для добавления дополнительных настроек, он включен в основной файл настройки `apache2.conf`

В файле `envvars` описаны переменные среды, необходимые для функционирования Apache-сервера.

В `ports.conf` вынесены настройки портов на которые можно будет подключиться к серверу или конкретному сайту на нем.

В папке `conf.d` находятся дополнительные конфигурационные файлы.

Для описания всех доступных сайтов используется папка `sites-available` в которой расположены файлы с описанием виртуальных хостов - `VirtualHosts`, опубликованные же сайты находятся в папке `sites-enabled` в виде ссылок на файлы доступных сайтов из папки `sites-available`.

Таким же образом в папках `mods-available` и `mods-enabled` настраивается доступность модулей используемых сервером.

Теперь необходимо подготовить компьютер к работе веб-сервера. Прежде всего необходимо создать единую папку для всех сайтов, которые будут там размещаться, например `/home/user/www`. Лучшее место для такой

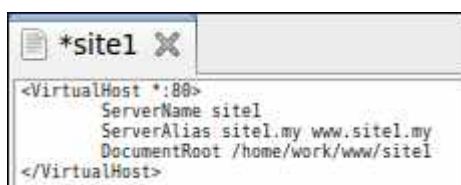
папки это домашний каталог пользователя. Далее в этой папке необходимо создать папку сайта. Например, /home/user/www/site1. И в эту папку кинуть файлы сайта.

Следующая команда (рис. 7) создает запись виртуального хостинга копируя стандартную запись из файла конфигурирования Apache:

```
work@work:~$ sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/site1
```

Рис. 7. Копирование файла конфигурации

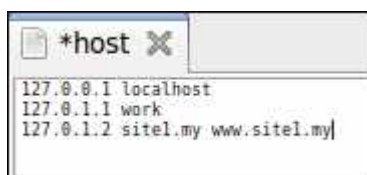
Теперь необходимо отредактировать файл, который находится по директории /etc/apache2/sites-available/site1. Необходимо настроить имя сервера, URL сервера и директорию, по которой находятся файлы сайта. После настроек файл конфигурации должен выглядеть, например, так, как на рис. 4.8.



```
*site1 X
<VirtualHost *:80>
  ServerName site1
  ServerAlias site1.my www.site1.my
  DocumentRoot /home/work/www/site1
</VirtualHost>
```

Рис. 4.8. Файл конфигурации сайта site1

Теперь необходимо как-то научить операционную систему распознавать домен .my. Для этого достаточно прописать необходимые строки в файле /etc/hosts, например так, как на рис. 4.9.



```
*host X
127.0.0.1 localhost
127.0.1.1 work
127.0.1.2 site1.my www.site1.my
```

Рис. 4.9. Редактирование файла hosts

Для начала необходимо разместить ссылку на VirtualHost в папку sites-enabled, и перечитать конфигурацию сервера Apache. Для создания ссылки можно выполнить такую команду и перечитать параметры (рис. 4.10). После этого ваш сайт, файлы которого размещаются в директории /home/user/www/site1 будет отображаться в браузере по адресу: site1.my или www.site1.my.

```
work@work:~$ sudo a2ensite sitel
Site sitel already enabled
work@work:~$ sudo /etc/init.d/apache2 reload
* Reloading web server config apache2
apache2: Could not reliably determine the serve
r's fully qualified domain name, using 127.0.1.
1 for ServerName
[ OK ]
```

Рис. 4.10. Активация сайта

Практическая работа

1. На виртуально машине разверните ftp-сервер;
2. Разрешите анонимный доступ для всех пользователей на данный ftp-сервер. Проверьте работу ftp-сервера с данной конфигурацией с вашей основной операционной системы;
3. Настройте разграничение прав доступа к определенным каталогам пользователей на ftp-сервере. Проверьте работу ftp-сервера с данной конфигурацией с вашей основной операционной системы;
4. Настройте смешанный режим доступа анонимных и зарегистрированных пользователей. Проверьте работу ftp-сервера с данной конфигурацией с вашей основной операционной системы;
5. Установить ssh-сервер на вашу операционную систему Linux. Настройте ssh с точки зрения безопасности;
6. На вашей основной операционной системе установить ssh-клиент (если основная операционная система Linux) или putty (если основная операционная система Windows). Проверьте работу ssh, настроив клиент соответствующим образом;
7. Установить веб сервер на Linux Ubuntu;
8. Создайте простую html-страничку. Разместите её на веб-сервере по веб-адресам: work.my и www.work.my.

Контрольные вопросы

1. Каково назначение ftp-сервера?
2. Каким образом производится настройка vsftpd?
3. Каково назначение сетевого протокола SSH?

4. Какие основные параметры рекомендуется менять при настройке SSH с точки зрения его безопасности и почему?

5. Каково назначение Telnet? Почему Telnet не рекомендуется использовать?

6. Каково назначение Apache?

7. Какие основные конфигурационные файлы Apache существуют?