Лабораторная работа 6. Установка и обзор ОС на базе GNU/Linux (Debian)
Теоретический материал.

Debian — операционная система, состоящая из свободного ПО с открытым исходным кодом. В настоящее время Debian GNU/Linux — один из самых популярных и важных дистрибутивов GNU/Linux, в первичной форме оказавший значительное влияние на развитие этого типа ОС в целом. Также существуют проект на основе другого ядра: Debian GNU/Hurd. Debian может использоваться в качестве операционной системы как для серверов, так и для рабочих станций.

Debian имеет наибольшее среди всех дистрибутивов хранилище пакетов — готовых к использованию программ и библиотек, — и если даже не по их числу, то по числу поддерживаемых архитектур: начиная с ARM, используемой во встраиваемых устройствах, наиболее популярных x86-64 и PowerPC, и заканчивая IBM S/390, используемой в мейнфреймах. Для работы с хранилищем разработаны разные средства, самое популярное из которых — Advanced Packaging Tool (APT).

Debian стал основой целого ряда дистрибутивов. Самые известные из них (в алфавитном порядке) — antiX, Knoppix, Linux Mint, Maemo, SteamOS, TAILS, Ubuntu.

Debian и дистрибутивы, основанные на нём (более 100), используют формат пакетов .deb и менеджер пакетов dpkg.

Выпуски

Выпуски Debian разделены на шесть веток:

- oldoldstable, содержащую пакеты предыдущего oldstable дистрибутива, является неофициальным LTS.
- oldstable, содержащую пакеты предыдущего стабильного дистрибутива; может одновременно являться неофициальным LTS и находиться в официальной поддержке после выхода Stable;
- стабильную (stable), содержащую пакеты, вошедшие в последний официальный дистрибутив (обновление пакетов в нём происходит только для устранения уязвимостей);

- тестируемую (testing), из которой будет формироваться следующий стабильный дистрибутив;
- нестабильную (unstable, sid), содержащую новые версии пакетов, которые готовятся к помещению в тестируемую ветку;
- экспериментальную (experimental), не являющуюся полноценной веткой — в ней находятся пакеты, требующие тщательного тестирования или которые повлекут серьёзные изменения в дистрибутиве.

Поддерживаемые архитектуры

Текущая стабильная версия официально портирована на следующие архитектуры:

- i386 — архитектура x86, разработана для Intel-совместимых 32-битных процессоров
- amd64 — архитектура x86-64 разработана для Intel/AMD 64-битных процессоров
- armel — архитектура ARM для Risc PC и различных встраиваемых систем
- armhf — архитектура ARM седьмой версии
- powerpc — архитектура PowerPC
- ia64 — архитектура Intel Itanium (IA-64)
- mipsel — архитектура MIPS с порядком байтов от младшего к старшему
- mips — архитектура MIPS с порядком байтов от старшего к младшему
- s390x — архитектура IBM System z
- arm64 — архитектура ARM, 64-бита (AArch64)
- ppc64el — архитектура Motorola/IBM PowerPC

Существуют также неофициальные версии для ряда других архитектур, некоторые из которых активно разрабатываются, но пока являются лишь частью нестабильного выпуска.

Настройка сетевых интерфейсов в Debian GNU/Linux

Для настройки сетевых интерфейсов в операционных системах, основанных на Debian, используется файл /etc/network/interfaces. Здесь должно находится описание для каждого интерфейса, способ получения IP адреса и другие параметры. В одном файле может быть настроено несколько интерфейсов.

Пример конфигурационного файла.

```
user@debian:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet dhcp
user@debian:~$
```

Описание каждого интерфейса, как правило, начинается со слова «auto» после которого идет имя интерфейса. Это означает, что указанный интерфейс должен быть запущен при выполнении ifup с опцией -a, именно она используется при инициализации системы. После строки «auto» идут другие строки настроек, которые касаются именно этого интерфейса.

Вместо параметра «auto», можно использовать «allow-hotplug», если нужно запускать интерфейс как только система обнаружит устройство.

Опции, которые применяются при настройке интерфейсов в файле /etc/network/interfaces:

- pre-up - выполнить команду перед запуском интерфейса;
- post-up - выполнить команду после запуска интерфейса;
- up - выполнить команду при запуске интерфейса;
- pre-down - команда перед отключением;
- post-down - команда после отключения;
- iface - указывает имя интерфейса;
- inet - указывает на использование стека TCP/IP, и IPv4;
- description - создать имя синоним для устройства;
- address - устанавливает ip адрес для статического соединения;

- netmask - установка маски сети;
- broadcast - широковещательный адрес;
- metric - приоритет для шлюза по умолчанию;
- gateway - шлюз по умолчанию;
- hwaddress - установить MAC адрес;
- mtu - размер одного пакета;
- и другие **.**

Примеры настроек сетевых интерфейсов

Использовать DHCP:

```
auto enp0s3
allow-hotplug enp0s3
iface enp0s3 inet dhcp
```

Ручная настройка:

```
auto enp0s3
iface enp0s3 inet static
address 192.168.0.7
netmask 255.255.255.0
gateway 192.168.0.254
```

Настройка виртуальных интерфейсов (несколько IP-адресов на одном физическом интерфейсе):

```
auto enp0s3
iface enp0s3 inet static
address 172.16.50.32
netmask 255.255.0.0
gateway 172.16.0.1
```

```
auto enp0s3:1
iface enp0s3:1 inet static
address 10.3.0.22
netmask 255.255.255.0
```

Настройка DNS

При отсутствии пакета resolvconf (отвечает за автоматическое назначение DNS серверов), настройки DNS определяются в файле /etc/resolv.conf

Пример файла:

```
domain example.com
search example.com
nameserver 8.8.8.8
nameserver 8.8.4.4
```

NetworkManager

На операционных системах с установленным графическим окружением для управления сетевыми подключениями, как правило, используется инструмент NetworkManager. В таких системах настройка сетевых интерфейсов производится в графическом меню, либо с помощью специальных консольных утилит (nmcli).

Протокол SSH

SSH (англ. Secure Shell — «безопасная оболочка») — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов). Схож по функциональности с протоколами Telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. SSH-клиенты и SSH-серверы доступны для большинства сетевых операционных систем.

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 (например, с веб-камеры). Также SSH может использовать сжатие передаваемых данных для последующего их шифрования, что удобно, например, для удалённого запуска клиентов X Window System.

Большинство хостинг-провайдеров за определённую плату предоставляет клиентам доступ к их домашнему каталогу по SSH. Это может быть удобно как для работы в командной строке, так и для удалённого запуска программ (в том числе графических приложений).

SSH — это протокол прикладного уровня. SSH-сервер обычно прослушивает соединения на TCP-порту 22. Спецификация протокола SSH-2 содержится в RFC 4251. Для аутентификации сервера в SSH используется протокол аутентификации сторон на основе алгоритмов электронно-цифровой подписи RSA или DSA, но допускается также аутентификация при помощи пароля (режим обратной совместимости с Telnet) и даже ip-адреса хоста (режим обратной совместимости с rlogin).

1. Аутентификация по паролю наиболее распространена. При каждом подключении подобно https вырабатывается общий секретный ключ для шифрования трафика.
2. При аутентификации по ключевой паре предварительно генерируется пара открытого и закрытого ключей для определённого пользователя. На машине, с которой требуется произвести подключение, хранится закрытый ключ, а на удалённой машине — открытый. Эти файлы не передаются при аутентификации, система лишь проверяет, что владелец открытого ключа также владеет и закрытым. При данном подходе, как правило, настраивается автоматический вход от имени конкретного пользователя в ОС.
3. Аутентификация по ip-адресу небезопасна, эту возможность чаще всего отключают.

Для создания общего секрета (сеансового ключа) используется алгоритм Диффи — Хеллмана (DH). Для шифрования передаваемых данных используется симметричное шифрование, алгоритмы AES, Blowfish или 3DES. Целостность передачи данных проверяется с помощью CRC32 в SSH1 или HMAC-SHA1/HMAC-MD5 в SSH2.

Список источников:

1. https://ru.wikipedia.org/wiki/Debian

2. https://losst.ru/nastrojka-seti-debian-9

3. https://wiki.debian.org/ru/NetworkConfiguration

4. https://ru.wikipedia.org/wiki/SSH