Расширенное администрирование Linux.

Блок 2

v 1.03

Оглавление

Служба точного времени

Принцип работы

Функциональность NTP основана на понятии главных серверов времени (называемых серверами первого эшелона), получающих сведения о точном времени из высокоточных источников, например от локально подключенной Глобальной системы рекогносцировки (GPS) или снимающих их с цезиевых часов.
Сервер, синхронизирующийся с сервером первого эшелона, называется сервером второго эшелона — эшелона исходного сервера + 1. По мере увеличения номера слоя точность времени может слегка снижаться. Принципиальными проблемами синхронизации времени являются учет сетевого ожидания и времени обработки пакетов и серверы с неточной установкой времени. Например, если сервер времени отправляет пакет «Точное время — 12:00:00, устано вите часы на 12:00:00», а пакету требуется 2 секунды на достижение места назначения, то часы на клиентском компьютере будут отставать на 2 секунды, Если на обработку пакета клиенту требуется еще 1 секунда, тогда клиентский компьютер будет отставать на 3 секунды.

NTP преодолевает эти проблемы несколькими способами:

- Измерением времени ожидания с помощью временных меток клиента и сервера;
- Учетом времени, необходимого на обработку сетевых пакетов;
- Использованием кратных выборок с множественных серверов для обеспечения точности;
- Составлением «черных списков» серверов, выдающих непоследовательные или неточные результаты.

NTP использует порт 123 UDP

В пакет входит следующее:

- *ntpq* для запроса серверов NTP;
- *ntpd* поддерживает точность локальных часов и (опционально) обеспечивает клиентам службу NTP;
- *ntptrace* прослеживает цепь сервера NTP к исходному серверу;
- *ntpdate* — одноразовая программа обновления часов.

Настройка на сервер точного времени

Для обеспечения большей точности часов сервера и снижения зависимости от доступности тех или иных серверов точного времени следует опрашивать пул серверов точного времени вместо одиночного сервера.

Онлайн список общедоступных серверов NTP
http://support.ntp.org/bin/view/Servers/WebHome

Синхронизация времени через NTP

Ручная синхронизация

```
ntpdate time.nist.gov ntp.ubuntu.com
```

```
18 Aug 17:32:35 ntpdate[3558]: step time server 80.127.4.179 offset -358.420872 sec
```

Установка времени:
```
ntpdate -bs ntp.ubuntu.com
```

Через Crontab
```
crontab –e
0 * * * * /usr/sbin/ntpdate [серверы NTP]
```

Троекратное упоминание сервера europe.pool.ntp.org говорит об использовании трех разных серверов, включенных в пул серверов времени.

Автоматическая синхронизация

Установка сервера:
(в /etc/apt/sources.list должен быть указан deb http://ftp.debian.org sarge main)

```
apt-get install ntp
```

Файл конфигурации:
```
/etc/ntp.conf
server ntp.ubuntu.com
server time.nist.gov
server europe.pool.ntp.org
```

Разрешение доступа из локальной сети:
По умолчанию ваш сервер NTP будет доступен всем хостам в Интернет. Параметр restrict в файле /etc/ntp.conf позволяет вам контролировать, какие машины могут обращаться к вашему серверу.
Если вы хотите запретить всем машинам обращаться к вашему серверу NTP, добавьте следующую строку в файл /etc/ntp.conf:
```
restrict default ignore
```

Если вы хотите разрешить синхронизировать свои часы с вашим сервером только машинам в вашей сети, но запретить им настраивать сервер или быть равноправными участниками синхронизации времени, то вместо указанной добавьте строчку

```
restrict 10.0.0.0 mask 255.0.0.0 nomodify notrap
```
/etc/ntp.conf может содержать несколько директив restrict
```
restrict 10.0.0.0 mask 255.0.0.0 noquery
```

Логи сервера:
```
/var/log/ntpstats/
```

Проверка запросов:
```
ntpq –p
```

Запуск сервера:
```
/etc/init.d/ntp start
```

FTP - протокол передачи файлов

1.Введение

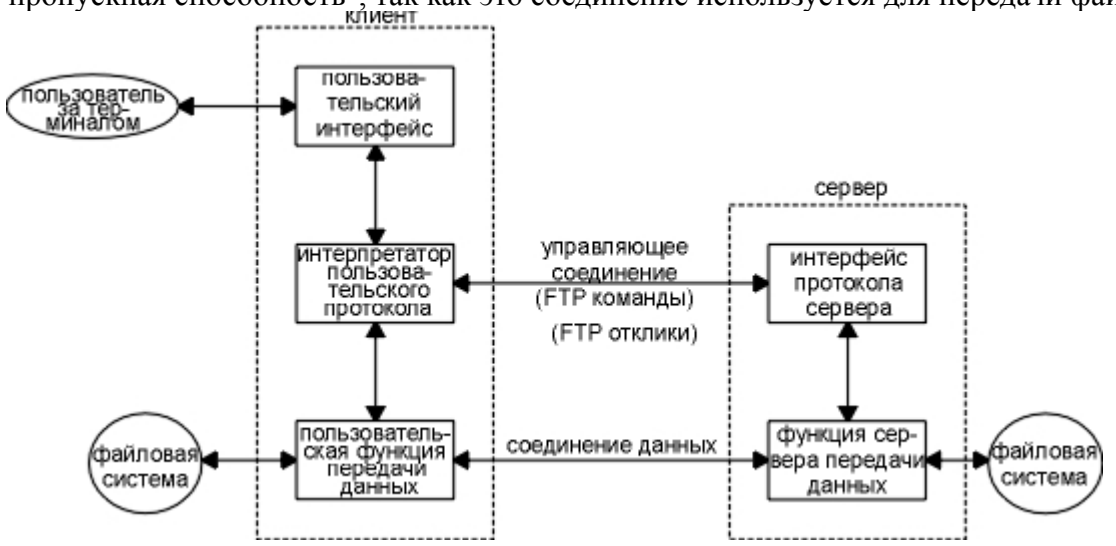
FTP является стандартом Internet для передачи файлов. Необходимо различать передачу файлов, именно то, что предоставляет FTP, и доступ к файлам, что предоставляется такими приложениями как NFS (Network File System, глава 29). Передача файлов заключается в копировании целого файла из одной системы в другую. Чтобы использовать FTP, необходимо иметь учетную запись (бюджет) на сервере, или можно воспользоваться так называемым анонимным FTP (anonymous FTP).

RFC 959 [Postel and Reynolds 1985] является официальной спецификацией FTP. Этот RFC описывает историю и развитиие передачи файлов в течение времени.

Протокол FTP

FTP отличается от других приложений тем, что он использует два TCP соединения для передачи файла.

1. Управляющее соединение устанавливается как обычное соединение клиент-сервер. Сервер осуществляет пассивное открытие на заранее известный порт FTP (21) и ожидает запроса на соединение от клиента. Клиент осуществляет активное открытие на TCP порт 21, чтобы установить управляющее соединение. Управляющее соединение существует все время, пока клиент общается с сервером. Это соединение используется для передачи команд от клиента к серверу и для передачи откликов от сервера.
2. Соединение данных открывается каждый раз, когда осуществляется передача файла между клиентом и сервером. (Оно также открывается и в другие моменты, как мы увидим позже.) Тип сервиса IP для соединения данных должен быть "максимальная пропускная способность", так как это соединение используется для передачи файлов.



Из рисунка видно, что интерактивный пользователь обычно не видит команды и отклики, которые передаются по управляющему соединению. Эти детали оставлены двум интерпретаторам протокола. Квадратик, помеченный как "пользовательский интерфейс", это именно то, что видит интерактивный пользователь (полноэкранный интерфейс, основанный на меню, командные строки и так далее). Интерфейс конвертирует ввод пользователя в FTP

команды, которые отправляются по управляющему соединению. Отклики, возвращаемые сервером по управляющему соединению, конвертируются в формат, удобный для пользователя.
FTP-сервер поддерживает 2 режима передачи данных: ascii и binary, что определяется переданными ему командами.

Команды FTP

Команды и отклики передаются по управляющему соединению между клиентом и сервером в формате NVT ASCII. В конце каждой строки команды или отклика присутствует пара CR, LF. Команды состоят из 3 или 4 байт, а именно из заглавных ASCII символов, некоторые с необязательными аргументами.

Команда*	**Описание**
help	получить список команд поддерживаемых ftp-сервером
ls или dir	список файлов или директорий
pwd	показать текущую директорию
cd	перейти к указанной директории
mkdir	создать директорию
rmdir	удалить директорию, если она не пустая
[m]get	получить файл[ы] с сервера
[m]put	отправить файл[ы] на сервер
TYPE {binary | ascii}	указать режим передачи данных
quit или exit	завершить работу с сервером

2. Устанавливаем ProFTPd и OpenSSL

Все команды выполняются от имени суперпользователя root, поэтому вам небходимо использовать sudo либо повысить свои привелегии командой:

```
sudo bash
```

Создаем нового пользователя test с паролем test:

```
adduser test
passwd test
```

Для установки ProFTPd и OpenSSL запустите

```
apt-get install proftpd openssl
```

Вам будет задан вопрос:

```
Запуск proftpd: <-- Самостоятельно
```

Из соображений безопасности вам необходимо добавить эти строки в /etc/proftpd/proftpd.conf

```
nano /etc/proftpd/proftpd.conf
DefaultRoot ~
IdentLookups off
ServerIdent on "FTP Server ready."
```

* В данной таблице указаны только те команды которые поддерживаются практически всеми ftp-серверами

3. Создаем SSL сертификат для TLS

TLS (англ. Transport Layer Security) — криптографический протокол, обеспечивающий защищённую передачу данных между узлами в сети Интернет. TLS-протокол основан на Netscape SSL-протоколе версии 3.0. Различие между SSL 3.0 и TLS 1.0 незначительные, поэтому далее в тексте термин «SSL» будет относиться к ним обоим.
TLS предоставляет возможности аутентификации и безопасной передачи данных через Интернет с использованием криптографических средств. Часто происходит лишь аутентификация сервера, в то время как клиент остается неаутентифицированным. Для взаимной аутентификации каждая из сторон должна поддерживать инфраструктуру открытого ключа (PKI), которая позволяет защитить клиент-серверные приложения от перехвата сообщений, редактирования существующих сообщений и создания поддельных.

SSL включает в себя три основных фазы:

- Диалог между сторонами, целью которого является выбор алгоритма шифрования
- Обмен ключами на основе криптосистем с открытым ключом или аутентификация на основе сертификатов.
- Передача данных, шифруемых при помощи симметричных алгоритмов шифрования

Для использования TLS нам необходимо создать SSL сертификат в каталоге /etc/proftpd/ssl

Создаем каталог /etc/proftpd/ssl

```
mkdir /etc/proftpd/ssl
```

Генерируем SSL сертификат

```
openssl req -new -x509 -days 365 -nodes -out  \
/etc/proftpd/ssl/proftpd.cert.pem -keyout \
/etc/proftpd/ssl/proftpd.key.pem
```

Вводим вашу регистрационную информацию

```
Country Name (2 letter code) [AU]: RU
State or Province Name (full name) [Some-State]: Moscow
Locality Name (eg, city) []: Moscow
Organization Name (eg, company) [Internet Widgits Pty Ltd]: CLASS
Organizational Unit Name (eg, section) []: IT
Common Name (eg, YOUR name) []: c230.unix.specialist.ru
Email Address []: root@localhost
```

4. Включаем TLS в ProFTPd

Для того, чтобы включить TLS для ProFTPd необходимо открыть файл конфигурации /etc/proftpd/proftpd.conf

```
nano /etc/proftpd/proftpd.conf
```

И раскомментировать строку

```
#
# This is used for FTPS connections
#
Include /etc/proftpd/tls.conf
```

Теперь откройте /etc/proftpd/tls.conf

```
nano /etc/proftpd/tls.conf
```

И отредактируйте его таким образом

```
TLSEngine on
TLSLog /var/log/proftpd/tls.log
TLSProtocol SSLv23
TLSOptions NoCertRequest
TLSRSACertificateFile /etc/proftpd/ssl/proftpd.cert.pem
TLSRSACertificateKeyFile /etc/proftpd/ssl/proftpd.key.pem
TLSVerifyClient off
TLSRequired on
```

Если у вас TLSRequired on, тогда только пользователи с включенным TLS получат доступ к вашему FTP серверу (могут возникнуть проблемы у пользователей использующих старые FTP клиенты не поддерживающие TLS). Для того чтобы все пользователи могли соединиться с FTP закомментируйте строку TLSRequired on, либо измените значение на Off

Перезапускаем ваш ProFTPd

```
/etc/init.d/proftpd restart
```

Теперь вы можете попробовать подключиться с использованием ftp-ssl клиента, или любого другого (если у вас TLSRequired off)

В случае возникновения проблем с TLS смотрите логи `/var/log/proftpd/tls.log`

5. Проверяем работу FTP-сервера

В силу того, что мы настроили ftp-сервер для работы через ssl, то следует установить ftp-клиент поддерживающий ssl.

```
aptitude install ftp-ssl
```

Теперь подключаемся к серверу соседа:

```
ftp-ssl IP_адрес_соседа
```

В качестве имени пользователя и пароля указываем test.

После подключения создаем на сервере директорию MyDir командой:

```
mkdir MyDir
```

Просим соседа убедиться в наличии директории.