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
```

Сетевой суперсервер

В современных дистрибутивах Linux вы можете встретить две разновидности программ, называемых сетевыми суперсерверами:

- inetd
- xinetd

Inetd является классическим вариантом программы. Xinetd — это дальнейшее развитие концепции сетевых суперсерверов UNIX систем.

После запуска сетевой суперсервер открывает на прослушивание порты, которые описаны в его конфигурационном файле /etc/inetd.conf или /etc/xinetd.conf. Если на порт приходит запрос на соединение, суперсервер запускает необходимую программу (согласно конфигурационного файла) и передает ей соединение.

Кроме этого xinetd:

- Позволяет контролировать доступ к службам TCP, UDP и RPC.
- Может ограничивать доступ к службам по времени.
- Имеет расширенные возможности журнальной регистрации.
- Имеет механизмы защиты от атак типа «отказ в обслуживании».
- Может использовать файлы /etc/hosts.allow и /etc/hosts.deny, а так же использовать демон tcpd.
- Привязать службы к определенным сетевым интерфейсам.

Программа inetd

Конфигурационный файл /etc/inetd.conf

- Сервис
- Тип соединения
- Протокол
- Флаги
- Пользователь
- Программа
- Аргументы

Программа inetd — это классическая разновидность сетевого суперсервера.

Конфигурационный файл программы — /etc/inetd.conf. На одну запись отводится одна строка. В строке может быть шесть или семь полей, в зависимости от ситуации. Поля разделяются пробелами или символами табуляции.

- Сервис — имя сервиса. Можно писать только сервисы, определенные в файле /etc/services.
- Тип соединения — возможные варианты: stream, dgram и raw.
- Протокол — имя транспортного протокола, используемого при соединении.
- Флаги — если для каждого соединения необходимо запускать новую программу,

значение флага nowait. Если одна запущенная программа может обработать несколько соединений — wait. В дополнении к параметру, можно написать число, определяющее максимальное количество подключений в минуту, например: nowait.30.
- Пользователь, с правами которого будет запущен данный процесс. Возможно указание группы. Например: user.group или user:group.
- Программа, которую запустит сетевой суперсервер. Если в этом поле стоит ключевое поле INTERNAL — inetd сам обработает та- кую службу и седьмое поле определять не надо.
- Параметры, которые будут переданы программе. Первый пара- метр — это обязательно имя самой программы.

Программа tcpd

Конфигурационные файлы:

/etc/hosts.allow

/etc/hosts.deny

В конфигурационном файле inetd.conf вместо имени программы у многих сервисов указывалась одна и таже программа — tcpd. Это, так называемый, tcp wrapper — программа, предназначенная для ограничения доступа к сервису.
У программы есть два конфигурационных файла:
- /etc/hosts.allow
- /etc/hosts.deny

В первом файле описано кому можно, во втором — кому нельзя, подключаться к сервисам. Файлы просматриваются в том же порядке, как они написаны выше.

Формат файлов одинаков:

```
программа : откуда [: параметры]
```

В первом поле указывается имя программы, а не имя сервиса. Во втором — откуда можно (hosts.allow) или откуда нельзя (hosts.deny) подключаться. В этом поле можно писать:
•IP адрес машины
•IP адрес сети с указанием маски подсети
•Имя машины. В именах можно использовать символ *.

Третье поле используется редко и не является обязательным.
В первом и втором полях можно использовать ключевые слова:
- ALL — все программы или хосты в сети.
- EXCEPT — за исключением перечисленных.

Программа xinetd

Конфигурационный файл /etc/xinetd.conf
Директория /etc/xinetd.d

В конфигурационном файле /etc/xinetd.conf описываются службы, доступ к которым

контролирует демон xinetd. Каждая служба описывается следующим образом:

```
service название_службы
{
     параметр оператор значение ...
     ...
}
```

Название службы должно быть описанно в файле /etc/services.
Кроме стандартных названий можно использовать и не стандартные, но в этом случае необходимо явно определять параметр port.

В таблице перечислены некоторые параметры, которые можно использовать при описании службы.

Параметр	***Описание***
socket_type	Тип соединения: stream (TCP), dgram (UDP), raw (неструктурированные пакеты), seqpacket (гарантированная последовательная доставка дейтаграмм).
protocol	Протокол по которому работает служба.
server	Полный путь к вызываемой программе.
server_args	Аргументы командной строки, которые следует передать программе при её вызове.
port	Если служба не описана в файле /etc/services, следует обязательно определить порт.
wait	Возможные значения: yes или no. В первом случае одна запускаемая программа может обслужить все соединения. Во втором случае на каждое соединение необходимо запускать свой экземпляр программы.
user	Определяет пользователя, с правами которого будет запускаться программа.
group	Определяет группу, с правами которой будет запускаться программа.
nice	Определяет значение nice запускаемой программы.
access_time	Время, когда можно обратиться к

Параметр	***Описание***
	службе. Время задаётся в виде чч:мм-чч:мм.
only_from	Список IP адресов или имен машин, разделенный пробелами, откуда можно подключаться к сервисам, обслуживаемым xinetd.
no_access	Список IP адресов или имен машин, разделен- ный пробелами, откуда нельзя подключаться к сер- висам, обслуживаемым xinetd.
instances	Целое число или ключевое слово UNLIMITED. Определяет максимальное количество одновремен- но запускаемых демонов.
log_type	“SYSLOG средство” - отсылать сообщения в систему syslog, используя определенное средство. “FILE файл [нижний предел [:верхний предел]]”- отсылать сообщения в файл, не используя систему xinetd. Предельные размеры файла указываются в килобайтах. При достижении нижнего предела, xinetd генерирует сообщение. При достижении верхнего – перестает записывать сообщения в файл.
redirect	Перенаправление соединения на другую машину в сети. В качестве параметра следует через пробел указывать IP адрес и порт.
bind	Определяет интерфейс, на котором сервер слушает запросы.
log_on_success	Определяет, какая информация в случае успешного соединения будет попадать в журнальные файлы. Возможные значения: PID, HOST, USERID, EXIT и DURATION.
log_on_failure	Определяет, какая информация будет попадать в журнальные файлы в случае неуспешной попытки установки соединения. Возможные значения: ATTEMPT, HOST, USERID, RECORD.
disable	Запрещает (значение равно yes) или разрешает (значение равно no)

Параметр	*Описание*
	службу. По умолчанию, доступ разрешён.
cps	Ограничивает количество подключений в единицу времени. Например cps = 25 30 – означает, что если в секунду будет больше чем 25 запросов на соединение, служба будет недоступна в течении 30 секунд.
max_load	Определяет максимальную загрузку сервера, при превышении которой xinitd перестаёт принимать запросы на соединение.
rlimit_*	Различные параметры, при помощи которых ограничиваются системные ресурсы, доступные программе. Например rlimit_as, rlimit_cpu, rlimit_data, rlimit_rss, rlimit_stack

При описании службы в конфигурационном файле необходимо указывать следующие обязательные атрибуты:

- socket_type
- wait
- user
- server
- protocol

В конфигурационном файле можно использовать параметр include, при помощи которого определяется директория, все файлы ко- торой будут подключены в основной конфигурационный файл.

Ниже приводится содержимое файла /etc/xinetd.conf:

```
defaults
{
     instances = 60
     log_type  = SYSLOG authpriv log_on_success   = HOST PID
     log_on_failure = HOST
     cps  = 25 30
}
includedir /etc/xinetd.d
```

defaults — определяет параметры по умолчанию для всех сервисов.

Дополнительные конфигурационные файлы находятся в директории /etc/xinetd.d, ниже приводится файла /etc/xinetd.d/pop3, описывающего сервис pop3.

```
service pop3
            {
                  socket_type     = stream
                  wait            = no
                  user            = root
                  server          = /usr/sbin/popa3d
                  log_on_success  += USERID
                  log_on_failure  += USERID
                  disable         = no
            }
```

Лабораторная работа

Цель работы

Научиться настраивать сетевой суперсервер xinetd.

Задачи

1. Убедитесь, что вы работаете с правами пользователя root.

2. Установите finger-сервер, следующей командой:
```
aptitude install xinetd fingerd
```

3. Создайте файл описания finger-сервера:
service finger
```
{
                socket_type   = stream
                wait     = no
                user     = root
                server   = /usr/sbin/in.fingerd
                log_on_success     += USERID
                log_on_failure     += USERID
                disable  = no
}
```

4. Заставьте сервер xinetd перечитать свои конфигурационные файлы:
```
/etc/init.d/xinetd force-reload
```

5. Убедитесь, что порты 79 открыт на прослушивание:
```
netstat -nlp | grep :79
```

6. Проверка работоспособности сервера finger:
```
finger root@ip_адрес_соседа
```
Вместо ip_адрес_соседа подставьте IP адрес соседа, работающего с Вами в паре.

7. Попытайтесь получить информацию от сервера finger вашего соседа. Затем от слушателя работающего в другой паре.

Вопросы

1. При помощи какого параметра можно указать сетевому суперсерверу, что он должен на каждое новое соединение запускать новый экземпляр программы?

2. Каким образом можно ограничить доступ с определенной машины к программе?

3. Вы разрешили использование службы. Каким образом заставить сетевой суперсервер перечитать свой конфигурационный файл?

FTP - протокол передачи файлов

1.Введение

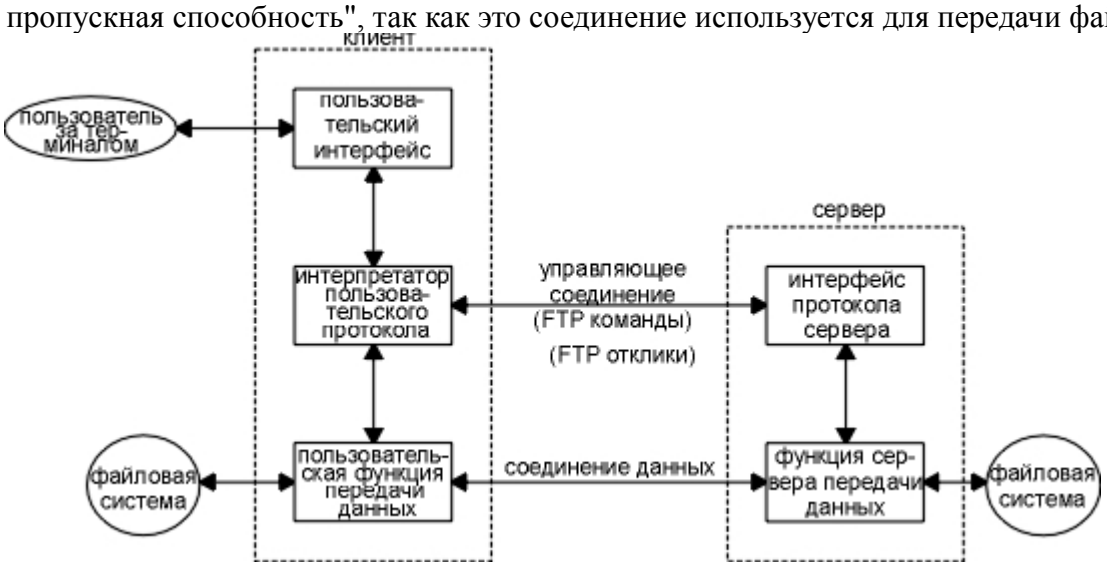
FTP является стандартом Internet для передачи файлов. Необходимо различать передачу файлов, именно то, что предоставляет FTP, и доступ к файлам, что предоставляется такими приложениями как NFS (Network File System, глава 29). Передача файлов заключается в копировании целого файла из одной системы в другую. 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.