Впервые опубликовано на developerWorks 02.09.2005

Подготовка к экзамену LPI 201: Предоставление доступа к файлам и сервисам

Intermediate Level Administration (LPIC-2) тема 209

Настройка сервера NFS

Использование NFS на стороне клиента

Если сервер правильно настроен, а у клиента есть соответствующие права, то монтирование удаленной файловой системы NFS потребует только использовать команду `mount`:

```
mount -t nfs my.nfs.server.com:/path/on/server /path/on/client
```

или же вставить соответствующую запись в /etc/fstab:

```
my.nfs.server.com:/path/on/server /path/on/client nfs rw,soft 0 0
```

Опция `soft` сообщает ядру, что надо посылать ошибку I/O (EIO) пользовательскому процессу в случае проблем с сетью. Опция по умолчанию `hard` вызовет зависание процесса, если сервер NFS недоступен.

В дополнение, программы помощники `rpc.lockd`, `rpc.statd` и `rpc.quotad` могут быть запущены на стороне клиента и/или сервера.

Настройка сервера NFS (часть первая)

Для полноценной работы сервера NFS требуется три различные программы, и ещё три могут быть запущены дополнительно.

Когда клиент NFS монтирует файловую систему NFS, он общается с демонами сервера, большинство из которых должны работать отдельно (а не быть запущены из `inetd`):

- `portmap`: Иногда именуется как `portmapper` или `rpc.bind`.
- `rpc.mountd`: Иногда `mounted`.
- `rpc.nfsd`: Иногда `nfsd`.

Дополнительно существуют три опциональные программы помощника:`rpc.lockd`, `rpc.statd`, и `rpc.quotad`, которые, соответственно, обеспечивают глобальное блокирование, ускорение семейства системных вызовов `lstat` (используется `ls -l` и другие), и обеспечивают поддержку квот.

Настройка сервера NFS (часть вторая)

Все три демона, относящиеся к NFS, используют TCP оболочки (`tcpd`) для управления доступом и поэтому могут потребовать записей в /etc/host.allow.

Ни `nfsd`, ни `portmap` обычно не требует никакой другой настройки кроме как /etc/hosts.allow.

Файлом конфигурации для `mountd` является (косвенно) /etc/exports. Он расписывает, какие файловые системы могут монтироваться определенными клиентами. В Linux реализации NFS, /etc/exports не обрабатывается прямо `mountd`. Вместо этого команда `exportfs -a` обрабатывает /etc/exports и затем пишет результат в /var/lib/nfs/xtab, который может быть прочитан `mountd`. Существуют другие флаги у `exportfs`, которые позволяют этим двум файлам быть рассинхронизированными. То есть, вы можете временно добавить или удалить экспортируемые каталоги, не модифицируя постоянных записей в /etc/exports.

Администраторам других Unix-подобных серверов следует принять во внимание, что синтаксис файла /etc/exports в Linux значительно отличается от того, что используется в SunOS или BSD.

Настройка /etc/hosts.allow и /etc/hosts.deny

Файл конфигурации /etc/hosts.allow описывает узлы, которые могут подключаться к системе Linux. Этот файл не относится к NFS, но системе необходимо разрешение на подключение к машине, чтобы она смогла использовать сервер NFS. Аналогично, файл /etc/hosts.deny это список узлов, которым запрещено подключаться.

Довольно неинтуитивно, сначала идет поиск разрешенных узлов, затем запрещенных узлов, но если машина не найдена, значит ей можно подключиться. Это не значит, что механизмы входа отдельных серверов нельзя изменять, внимательный администратор может запретить все подключения, которые не разрешены явно (немного паранойи не помешает) с помощью:

```
# /etc/hosts.deny
ALL:ALL EXCEPT localhost:DENY
```

Установив запрет в /etc/hosts.deny на все (кроме соединений из LOCALHOST) получаем, что возможны только указанные явно соединения. Например:

```
#/etc/hosts.allow
# Allow localhost and intra-net domain to use all servers
ALL : 127.0.0.1, 192.168.
# Let everyone ssh here except 216.73.92.* and .microsoft.com
sshd: ALL EXCEPT 216.73.92. .microsoft.com : ALLOW
# Let users in the *.example.net domain ftp in
ftpd: .example.net
```

Настройка /etc/exports

Вот простой пример файла /etc/export:

```
# sample /etc/exports file / master(rw) trusty(rw,no_root_squash)
/projects proj*.local.domain(rw) /usr *.local.domain(ro) @trusted(rw)
/home/joe pc001(rw,all_squash,anonuid=150,anongid=100) /pub
(ro,insecure,all_squash)
```

Обычно, `root (uid 0)` на стороне клиента выглядит как nobody (uid 65534) на стороне сервера; это называется смещением root так как оно позволяет защитить файлы, владельцем которых является root (а члены группы/остальные не имеют права на запись) от изменений клиентами NFS. Опция `no_root_squash` отменяет такое поведение, и позволяет пользователю root иметь довереный полный доступ к разделу/. Такой доступ может быть полезен при установке и настройке программного обеспечения.

Раздел `/usr` может быть только прочитан всеми узлами, кроме тех, что находятся в сетевой группе "trusted" .

Когда `/home/joe` монтируется pc001, все удаленные пользователи (в независимости от uid/gid) будут иметь uid=150, gid=100. Это может быть полезно, если удаленный клиент NFS представляет собой однопользовательскую рабочую станцию или не поддерживает несколько пользователей (как в случае с DOS).

Обычно, Linux (и другие Unix-подобные операционные системы) резервируют TCP и UDP порты от 1-1023 (так называемыебезопасные порты) для использования процессами пользователя root. Чтобы удостовериться, что именно root инициировал удаленное подключение NFS, сервер NFS обычно требует, чтобы удаленные клиенты использовали безопасные порты. Это соглашение, однако, не соблюдается некоторыми операционными системами (например Windows). В таких случаях опция insecure позволяет клиенту NFS использовать любой порт TCP/UDP. Обычно она требуется при обслуживании клиентов Windows.

Утилиты NFS

`nfsstat` отображает статистику NFS (клиента и/или сервера) подобно тому, как на локальной машине эти делают утилиты `iostat` и `vmstat`.

Команда `showmount` запрашивает у `mountd` и показывает, какие клиенты в настоящий момент смонтировали файловые системы. NFS это протокол, не использующий информацию о состоянии и обращение к демону `mountd` происходит нечасто, поэтому вывод `showmount` может быть некорректным. К сожалению, нет способа заставить `showmount` быть точным. Однако, неточность `showmount` заключается лишь в том, что она всегда показывает устаревшие записи.

В этом контексте "не использующий информацию о состоянии" означает, что демон `nfsd`, который работает с реальной информацией, не помнит о том, ни какие файлы были открыты ни какие клиенты смонтировали какие разделы. Каждый запрос

(readblock, writeblock и другие) содержит всю необходимую информацию для его выполнения (номер раздела, который предоставляется mountd, inode номер, номер блока, read/write/ и другие данные). Протокол HTTP очень похож в этом отношении. Другой стороной не использования информации о состоянии является то, что если сервер перезагрузится, то клиенты заметят только небольшой период времени потери доступа к системе.

Настройка сервера samba

Настройка Samba сервера

Сервер Samba smbd обеспечивает доступ к файлам и принтерам (в основном для клиентов Windows). В то время как он может быть запущен из inetd, обычно он запускается как отдельный демон smbd -D. Nmbd это сервер имен NetBios (или WINS сервер). Он тоже может запускаться из inetd, но обычно его запускают отдельно nmbd -D. Samba может работать как сервер в рабочей группе Windows, а также как главный контроллер домена.

Файлом настройки для smbd и nmbd является /etc/samba/smb.conf. Огромное количество параметров описано в man-странице smb.conf. Файл lmhosts используется для отображения имен NetBios на IP адреса. Его формат схож с (но не идентичен) файлом /etc/hosts.

Есть несколько великолепных HOWTO и книг о том, как настраивать Samba. Этот раздел затрагивает несколько основных идей с указанием, где можно найти больше информации.

Предоставление доступа к домашнему каталогу

Следующий фрагмент smb.conf позволит пользователям получать доступ к домашним (локальным) каталогам с удаленных клиентов Samba:

```
[homes]
comment = Home Directories
browseable = no
```

Обычно эти строки включены по умолчанию в smb.conf.

Разделение доступа к принтеру с помощью CUPS

Из многочисленных систем печати Unix CUPS одна из самых старых и, возможно, наиболее популярная. В зависимости от вашего дистрибутива, CUPS может как включена, так и отключена по умолчанию в smb.conf. Вот простой пример предоставления совместного доступа к принтеру с помощью CUPS:

```
[global]
       load printers = yes
       printing = cups
       printcap name = cups
```

```
[printers]
  comment = All Printers
  path = /var/spool/samba
  browseable = no
  public = yes
  guest ok = yes
  writable = no
  printable = yes
  printer admin = root

[print$]
  comment = Printer Drivers
  path = /etc/samba/drivers
  browseable = yes
  guest ok = no
  read only = yes
  write list = root
```

CUPS может предоставить ppd (Postscript printer description) файлы и драйверы Windows клиентам, которые, будучи правильно установлены, позволят удаленным пользователям использовать весь список возможностей принтеров (выбор цветной или черно-белой печати, разрешения, выбор размера бумаги, двусторонней или односторонней печати и так далее). Традиционные системы печати Unix довольно неудобны. Смотрите man-страницу cupsaddsmb для получения более подробной информации.

Аутентификация

Samba (в отличие от NFS) требует, чтобы каждый пользователь прошел аутентификацию. Как и с другими сетевыми сервисами, требующими аутентификацию, следует удостовериться, что пароли не передаются по сети незашифрованными. Смотри раздел шифрование паролей в man-странице smb.conf.

Существует несколько механизмов, с помощью которых Samba может аутентифицировать удаленных пользователей (клиентов). По своей природе, большинство из них несовместимо со стандартным хэшем паролей Unix. Никогда не передавайте пароли по сети в открытом виде. Это всегда плохая идея.

Предположим, что вы шифруете пароли в сети, `smbpasswd` обычно будет использоваться для установки пользователям начальных паролей Samba. Опция "Unix password sync" позволит `smbpasswd` изменить пароли Unix, когда пользователи меняют свои пароли Samba.

В другом случае настроенный модуль `pam_smb` может аутентифицировать пользователей Linux используя базу данных Samba. Если этого недостаточно, то можно настроить LDAP для аутентификации пользователей Samba и/или Linux.

Отладка Samba

При настройке сервера Samba довольно полезной может оказаться команда

`testparm` (также называемая `smbtestparm`). Она просмотрит smb.conf и сообщит об ошибках.

Команда `nmblookup` делает для Samba тоже, что и `nslookup` для DNS; она делает запрос справочника NetBios. На `man`-странице `nmblookup` можно узнать подробности.

Настройка клиентов Samba

Команда `smbclient` предоставляет доступ к общим папкам Samba. Прозрачный доступ к папкам SMB довольно непрост; на man-странице `smbmount` можно узнать подробности.

Настройка сервера File Transfer Protocol

Об FTP

FTP это старый и широко используемый сетевой протокол. FTP обычно работает на двух раздельных портах -- 20 и 21. Порт 21 используется для контрольного потока (передающий информацию о входе в систему и команды) в то время как порт 20 используется для потока данных, по которому идет передача файлов.

В общем FTP это не очень безопасный протокол по той причине, что в режиме по умолчанию управляющий поток -- а значит логин и пароль -- передается в открытом виде. Поток данных также передается открыто, также как в NFS и Samba (для усиления безопасности SSH/SCP лучший выбор). Можно перенаправить управляющий поток FTP через SSH, таким образом, защитив его.

Традиционные клиенты FTP предоставляют собственные среды для работы, по которым передаются команды и настраиваются соединения. Иногда GUI интерфейсы поставляются вместе с клиентами для более удобной работы с файлами. Однако в наши дни многие инструменты включают FTP -- начиная от менеджера файлов до текстовых редакторов, они часто работают с файлами, лежащими на FTP сервере.

Анонимный FTP

Для тех применений, где FTP используется чаще всего, безопасность не является проблемой. Чаще всего FTP сервера используются как "анонимные FTP" -- то есть размещенные на них данные доступны миру и не требуют большой безопасности. По соглашению пользователь anonymous может получить доступ к файлам, предоставив произвольный пароль (обычно адрес почты), который не проверяется. Иногда имя пользователя/пароль требуются, полученная комбинация не проходит более сложную степень аутентификации (например для людей, которые хотят стать добровольцами в каком-либо проекте).

Большинство Web-браузеров и файловых менеджеров, а также инструментов прозрачно поддерживают FTP сервера. Часто этим инструментам требуется FTP URL для получения файла (а также для загрузки файла на сервер). Например,

инструмент командной строки wget скачает файл с FTP сервера с помощью следующей команды:

```
$ wget ftp://example.net/pub/somefile
$ wget ftp://user:passwd@example.net/pub/somefile
```

Файловые менеджеры часто монтируют FTP сервера как локальную систему или диски NFS или Samba (но не полностью точно так же, не используйте mount и /etc/fstab; такие псевдоразделы обычно именуются по своему URL).

Выбор FTP серверов

FTP старый и повсеместно внедренный, поэтому существует огромное число его реализаций и установок на различных Linux дистрибутивах. Настройка выбранного FTP сервера потребует от вас обращения к руководству по установке.

Некоторые популярные Linux FTP сервера:

- wu-ftpd.
- vsftpd.
- ProFTPd.
- BSD ftpd.
- TUX FTP.

Многие другие реализации не так популярны. В большинстве случаев настройка конкретного сервера будет находится в файле /etc/FOOftpd.conf (для соответствующего сервера "FOO"). Мне нравится vsftpd, так как он быстр и довольно надежен в плане защищенности ("vs" означает "очень защищенный").

Простой пример настройки FTPd

Синтаксис настройки каждого сервера будет различен. Но некоторые концепции, взятые из /etc/vsftpd.conf позволят понять типы опций остальных серверов. Что касается `vsftpd`, то у каждой опции имеется формат вида `option=value` с использованием знака решетки для обозначения комментариев. Большинство остальных файлов настроек FTPd похожи.

- `anonymous_enable`: управляет возможностью входа пользователя anonymous.
- `anon_world_readable_only`: Когда включено, то только anonymous пользователям позволено скачивать файлы для чтения.
- `chroot_local_user`: Если включено, то локальные пользователи будут помещены в chroot() окружение в своих домашних каталогах после входа.
- `pasv_enable`: Должен ли сервер использовать "пассивный FTP" режим, в котором клиенты инициируют порты (помогает, когда у клиентов есть файервол).
- `ssl_enable`: Если включено, то `vsftpd` будет поддерживать SSL соединения.
- `tcp_wrappers`: Если включено, то все входящие соединения будут проходить контроль доступа (как /etc/hosts.allow и /etc/hosts.deny).

Запуск FTP сервера

В простейшем случае вы можете запустить FTP сервер, так же как и любой другой демон:

```
% sudo vsftpd
```

Теперь сервер будет принимать входящие соединения согласно правилам в его файле настроек. Вы также можете запустить FTP сервер из "сетевого супер-сервера", такого как `inetd` или `xinetd`. Руководства LPI 202 расскажут о супер-серверах.

Запуск демона отдельно, даже в скриптах загрузки, как для какого-то определенного уровня загрузки, так и в /etc/rcS.d/, даст вам точный контроль над поведением FTP сервера.