

# Администрирование Linux

## Лекция 9

### Сетевые файловые системы

Иртегов Д.В.

Новосибирский гос. Университет

2014

# Основные сетевые ФС, поддерживаемые Linux

- NFS
  - Поддерживает все операции и атрибуты файлов и каталогов, требуемые стандартом POSIX
  - Можно размещать домашние каталоги, исполняемые файлы и разделяемые библиотеки
- SMBFS (CIFS, Microsoft Network)
  - Удобна для обмена файлами с DOS/OS/2/Windows
  - НЕ поддерживает стандартные атрибуты POSIX
  - Есть бесплатный сервер Samba, позволяющий использовать Linux в качестве файлового сервера и контроллера домена NT4 (изучать не будем)

# Демонстрация

- Монтирование сетевого ресурса \\10.4.16.10\USER через smbfs

# Сетевая файловая система NFS

- Network File System,
- Первоначально разработана Sun Microsystems во второй половине 1980х
- NFS v1, v2
  - Stateless (попытка защититься от перезагрузок сервера)
  - Собственный транспортный протокол поверх UDP
  - Динамический выбор порта UDP (portmap/rpcbind)
  - Сервер в виде модуля ядра + вспомогательные userland сервисы
  - Аутентификация по IP/UID, позднее NIS+/DES, Kerberos

# NFS (продолжение)

- NFS v3
  - Не полностью stateless
  - Расширения для блокировки файлов (lockd)
  - Поддержка UDP и TCP
  - Фиксированный порт TCP  
возможна работа через firewall

# NFS (продолжение)

- NFS v4
  - lockd/mountd стали частью основного протокола
  - Только TCP, по умолчанию порт 2049
  - Аутентификация не по UID, а по имени user@host
  - Нужен дополнительный сервис idmap, преобразующий имена в локальные UID

# Немного про UDP vs TCP

- Первоначально NFS был рассчитан на работу в локальных сетях, соединенных коаксиальным кабелем или 10BaseT хабом.
- В этих условиях, пакеты почти не теряются (сетевая карта получает сигнал о коллизии и сама переотправляет пакет)
- В сетях с коммутаторами и роутерами, перегрузка сети ведет к потере пакетов
- NFS/UDP при потере пакета переотправляет весь запрос, что усугубляет перегрузку сети
- TCP гораздо умнее
- UDP может давать выигрыш на тестах в слабо загруженной сети, но в реальных сценариях рекомендуют использовать TCP
- UDP оставлен только для совместимости

# Еще общие слова про NFS

- Легко развернуть, сложно планировать
- Главная проблема – безопасность
- UID легко подделать, поэтому аутентификация по UID возможна только в полностью доверяемой сети
- Рекомендованное решение – Kerberos
- Возможные альтернативы
  - Аутентификация 802.1x + static ARP
  - VPN для каждого клиента

# Режимы использования NFS

- Синхронизация UID и имен
  - NIS/NIS+
  - LDAP
- Без синхронизации UID
  - all\_squash – все клиентские UID отображаются на anonuid
  - Удобно для Readonly файловых систем

# Демонстрация

- Утилита `showmount`
- Монтирование каталога  
`10.4.16.156:/export/musom`
- Монтирование каталога  
`10.4.16.156:/export/home/vpupkin` на  
`/home/vpupkin`
- Графический вход пользователем  
`VPentyshkin`

# Упражнение

- Делать по очереди!!!
- Монтирование `/export/home/vpupkin`
- Вход пользователем `VPentyshkin`

# Введение в automount

- Монтировать каждый домашний каталог отдельно – скучно
- Монтировать весь /export/home негибко (не получится держать разных пользователей на разных серверах)
- Монтировать из /etc/fstab – проблемы (некоторые юниксы падают в single user, когда не могут смонтировать запись из /etc/fstab)
- Куды крестьянину податься?

# autofs

- Автоматическое монтирование файловых систем при первом обращении
- Автоматическое размонтирование по неактивности
- Конфигурация:
  - `/etc/auto.master`
  - `/etc/auto.home` (файл)
  - `/etc/auto.net` (скрипт, пример)
  - Карты NIS, LDAP (не будем изучать)

# Демонстрация

- Отсутствие домашних каталогов пользователей test2, test3 в /home
- `ls -la /home/test2`
- `mount`
- Проблема: если сделать автоматическое монтирование /home, ваш домашний каталог тоже пропадет
- Его надо переносить на NFS

# Упражнение

- Отредактировать `/etc/auto.master`
  - Включить строку  
`/home /etc/auto.home`
- Скопировать `/etc/auto.home` из `10.4.16.156:/export/home/auto.home`
- Скопировать ваш домашний каталог из `/home` на `10.4.16.156`
  - `mkdir /nfsmnt; mount 10.4.16.156:/export/home /nfsmnt; cp -rp /home/$USER /nfsmnt`
  - Добавить строчку  
`$USER 10.4.16.156:/export/home/$USER`  
в `/etc/auto.home`
- Рестартовать `autofs`
  - `service autofs restart`
  - Посмотреть, что произошло с вашим домашним каталогом
  - На всякий случай, перелогиниться

# Настройка сервера nfs

- Ставится пакетом `nfs-utils`
- Стартует сервисом `nfsd`
- на самом деле, поднимает еще несколько зависимых сервисов: `lockd`, `mountd`, `rpcbind`, `idmap`
- Главный файл конфигурации: `/etc/exports`
- Команда `exportfs`

# Демонстрация

- `/etc/export` на сервере `10.4.16.156`
- Демонстрация, что `root` может создавать файлы и каталоги через `nfs`
- Изменение параметра `root_squash` у шары `/export/home`
- Демонстрация, что `root` теперь НЕ может создавать файлы чере `nfs`

# Обзор параметров шар NFS

- Что: имя каталога на сервере
- Кому: hostname, маска, IP адрес, IP сеть
- Как: (опции)
  - ro, rw
  - root\_squash
  - all\_squash
  - mountpoint=path  
шарить только если смонтировано
  - sec=sys,krb5

# Обзор параметров клиента NFS

- Стандартные опции монтирования
  - ro, noexec, nosuid
- NFS-only
  - nfsver=3,4
  - bg, fg  
фоновое монтирование, если сервер не отвечает
  - soft, hard, intr  
повторять ли попытки, если  
[есть подозрение что] сервер отвалился
  - tcp, udp

# Упражнение

- Расшарить каталог  
/opt 10.4.16.0/24(rw, root\_squash)
- Создать там каталог, доступный на запись одному из пользователей LDAP
- Смонтировать соответствующие каталоги других студентов в /mnt по протоколу nfs3
- Убедиться, что вы можете создавать файлы из-под пользователей LDAP, и все их видят

# Продолжение упражнения

- Настроить `idmap.conf` для отображения пользователей через `Idap` (информацию взять из `/etc/sss/sss.conf`)
- Перемонтировать каталоги по протоколу `nfs4`
- Настроить отображение пользователей с `uid=500` на вашего локального пользователя с тем же `uid`, через секцию `[Static]` в `idmapd.conf`
- Создать каталог, доступный этому пользователю.
- Убедиться, что пользователи ваших станций с `uid=500` могут работать с этим каталогом

# Применение NFS

- Домашние каталоги пользователей
- Разделяемые данные в вычислительных кластерах
- Файлопомойки
- Каталоги `/bin`, `/lib`, `/usr` – удобно централизованно устанавливать апдейты
- На бездисковых станциях – корневая ФС (сейчас чаще используют iSCSI)