/conf

RootConf – 2009

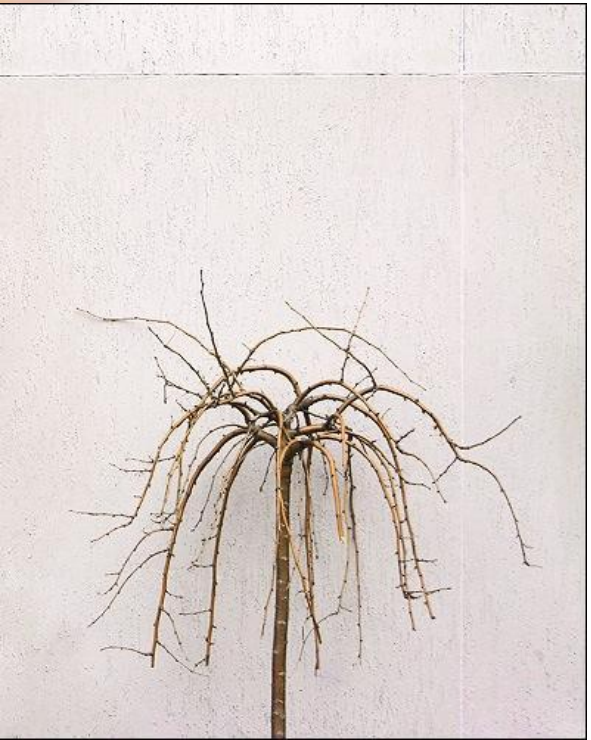
Профессиональная конференция системных администраторов

Использование LDAP в инфраструктуре интернет-компании

Сергей Скворцов

2009-04-14

$Revision:: 45 $



I. Теория

Служба каталогов

- Служба каталогов (Directory Service)
  - Централизованное (единое) хранилище
  - Хранит ресурсы – один или более классов, есть набор свойств (атрибутов)
  - Иерархическое (tree) представление

Что такое LDAP?

- **L**ightweight **D**irectory **A**ccess **P**rotocol
  - *Облегчённый* вариант DAP (входит в X.500)
  - Кратенько: RFC 4510..4533, плюс ещё ~10..20
  - Но это не страшно!
    - Кто читал весь стандарт SQL?
    - А всё RFC по DNS?
  - Это бинарный протокол (TCP тоже такой :)

Глобальные сущности

- **D**irectory **I**nformation **M**odel (DIM)
  - Набор схем, которые описывают классы и атрибуты
- **D**irectory **I**nformation **T**ree (DIT)
  - Собственно хранилище ресурсов (directory entries) – объектов и aliases (грубо: symlinks)

Концепции: классы

- Каждый объект относится как минимум к одному классу. Классы наследуются.
- Два вида:
  - Структурные (structural)
  - Вспомогательные (auxiliary)
- Содержат обязательные (MUST) и вспомогательные (MAY) атрибуты

Пример: класс

```
objectclass (
    0.9.2342.19200300.100.4.5
    NAME 'account'
    SUP top STRUCTURAL
    MUST userid
    MAY ( description $ seeAlso $
          localityName $ organizationName $
          organizationalUnitName $ host
        )     )
```

Концепции: атрибуты

- Атрибут определяется через:
  - Имя и OID-идентификатор
  - Синтаксис (syntax)
  - Правила (matching rules)
    - Сравнения, поиска подстроки
    - Упорядочивания (сортировки)
  - Тип значения: множественное или единственное (SINGLE)

Пример: атрибут

```
attributetype ( 2.5.4.9
  NAME ( 'street' 'streetAddress' )
  DESC 'RFC2256: street address of object'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128}
)
```

Схема(-ы)

- Схема задаёт набор атрибутов и классов
- Набор загружаемых файлов схем образуют схему DIT (или просто DIM).
- Схемы есть:
  - Предопределённые (системные)
  - Внешние (application)
  - Ваши собственные

И снова про DS

- Структурированное, иерархическое хранилище
- Оптимизация на массовое чтение, редкие записи
- Это НЕ реляционная модель!
- Может быть разбито на поддеревья
- Рассчитано на распределённое использование (referrals)

DS на пальцах

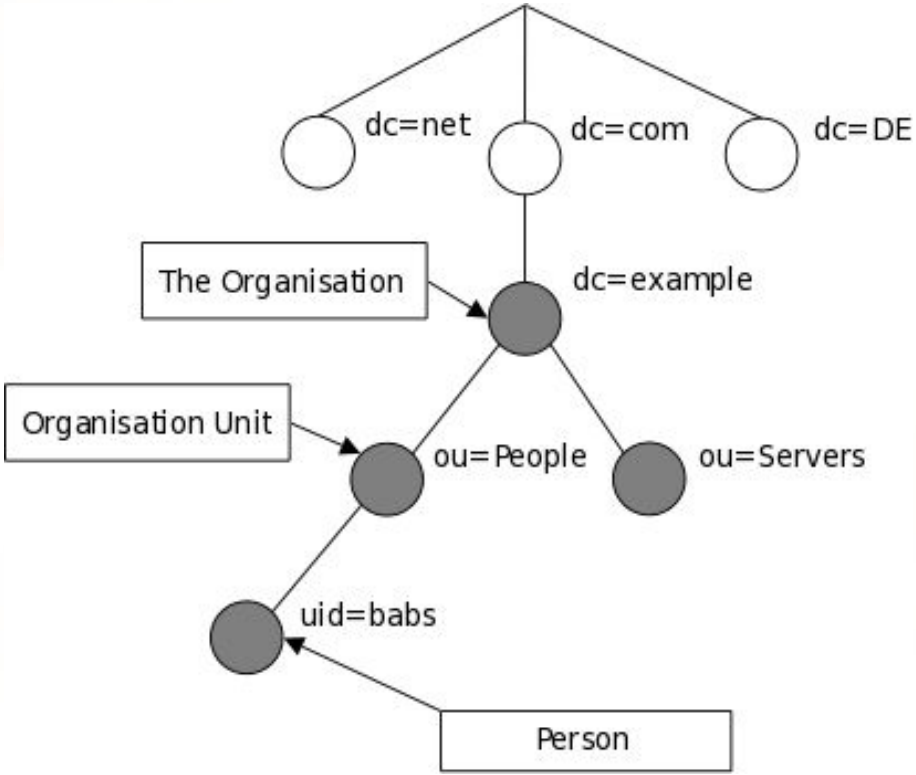
- Аналог – DNS
  - Но! DNS – это простая lookup служба
  - Служба каталогов – куда круче!
- Некоторая симметрия
  - Реплики и поддеревья – зоны
  - DNS записи как объекты
  - Master и slave сервера
  - И т.п.

Кусочек дерева

```
o=company
   ou=users
      cn=vpupkin
      ...

DN: cn=vpupkin,ou=users,o=company
```

Картинка


dc=net
dc=com
dc=DE
The Organisation
dc=example
Organisation Unit
ou=People
ou=Servers
uid=babs
Person

Кусочек дерева

```
o=company
   ou=users
      cn=vpupkin
      ...

DN:  cn=vpupkin,ou=users,o=company
```

LDIF – текстовый формат

```
dn: uid=vpupkin,ou=users,o=company
objectClass: person
objectClass: inetLocalMainRecipient
cn: Vasily Pupkin
userPassword: {SSHA}XXXXXX
mail: vasya@pupkin.ru
```

Схема расширяема

- Решить, надо ли вам это
- Достаточно:
  - Прочесть раздел «Extending Schema»
  - RFC 4512, и немного других :)
- Зарегистрировать OID в IANA

Стандартные классы

Системные:

organization organizationalUnit groupOfUniqueNames

Пользователи:

account inetOrgPerson inetLocalMailRecipient

Сеть:

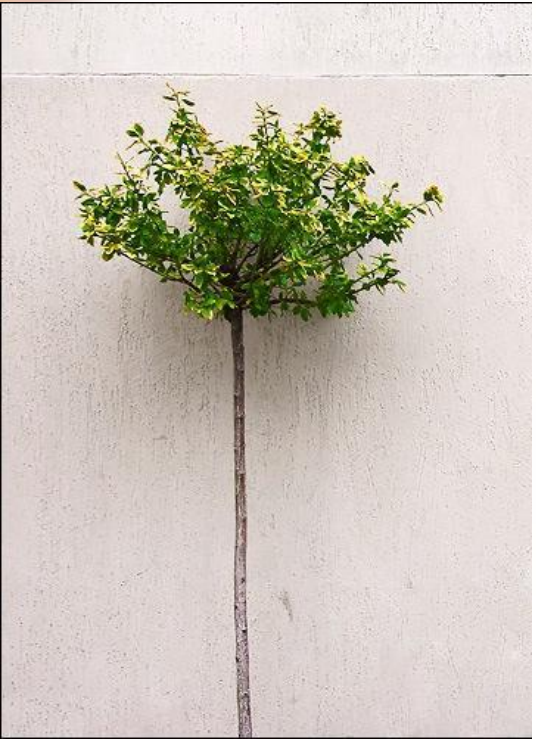
domain ipHost ipNetwork bootableDevice ieee802Device

UNIX-related:

posixAccount posixGroup sudoRole ldapPublicKey

Что ещё важного?

- Поиск – через т.н. фильтры
  - Достаточно мощный язык запросов
- Расширения протокола через controls
  - Примеры: Paged Results, Modify Password
- Безопасность
  - TLS, SASL, и т.п.



I I. Практика

Поставленные цели

- Единое центральное хранилище
- Управление учётными записями сотрудников
- Управление почтой (aliases, листы рассылки)
- Учёт сетевых ресурсов (сетей, хостов)
- Единая, прозрачная аутентификация и авторизация
- Конфигурация (в т.ч. ACL) для

Цель: управление доступом

- Единая точка управления доступом:
  - К host'ам по SSH; управление sudoers
  - К веб-ресурсам: intranet, wiki, bugzilla, otrs, SVN, webdav-folders, внутренние приложения…
  - К прочим ресурсам: SMTP/IMAP/POP3; KVM / IPMI, …
  - К почтовым рассылкам (в т.ч. архивам – через IMAP shared folders)

Цель: инвентаризация

- Отправная точка для инвентаризации:
  - Информация о хостах, сетях
  - Ссылки во внешние системы учёта
- Точка синхронизации с DNS:
  - DNS-зоны доменов генерятся *частично* из LDAP
- Мониторинг:
  - Начальную информацию берёт из LDAP

Выбор software

- OpenLDAP
  - Стабильный, быстрый, функциональный
  - Opensource
- Альтернативы
  - Novell eDirectory; Microsoft Active Directory
  - Apache Directory Server; Fedora Directory Server
  - И ещё чуть меньше десятка

Реализация: сервера

- Master-серверы:
  - FreeBSD 7.1; OpenLDAP 2.4.16
  - 2 сервера в MirrorMode
  - CARPed
- Consumer-серверы в отдельных ДЦ
  - чтение локально
  - записи перенаправляются на masters

Типичный LDAPified host

- OS: *nix (у нас в массе FreeBSD)
- pam_ldap – PAM смотрит сначала в LDAP:
- nss_ldap – интеграция NSS с LDAP
  - users, groups, passwd
- openssh+LPK – SSH-ключ смотрится в атрибуте sshPublicKey учётной записи пользователя
- sudo+ldap – sudoers полностью находится в LDAP; локальный sudoers у нас запрещён

DIT крупным планом

```
cn=company-log
o=company
   ou=users
   ou=groups
   ou=mail
   ou=networks
   ou=sites
```

Учётные записи: DIT

```
o=company
   ou=users
      cn=vpupkin
      ...
      ou=retired
      ou=locked
      ou=system
```

Учётные записи: типы

- ext-user
  - person, inetOrgPerson, inetLocalMailRecipient, extraPerson
    - `uid, cn, givenName, sn, mail, userPassword, icqNumber, jabber, birthday`
- plain-user
  - + organizationalPerson, intraPerson
    - title, roomNumber, telephoneNumber, shirtSize
- account-user
  - + posixAccount, shadowAccount, ldapPublicKey, intraAccount
    - uidNumber, gidNumber, loginShell, sshPublicKey, loginClass

Почта: DIT

```
o=company
  ou=mail
    dc=spylog.ru
      ou=aliases
        cn=mOrketing
      ou=lists
        cn=changes
            cn=read
            cn=write
```

Почта: пример

```
dn: cn=changes, ou=lists,
  dc=spylog.ru,ou=mail,o=company

objectClass: nisMailAlias
rfc822MailMember: vasya@pupkin.ru
owner: uid=vpupkin,ou=users,o=company
```

LDAP + SMTP

- SMTP-auth пользователей – через LDAP
- Exim, при получении письма для домена dc=XXX,ou=mail,обрабатывает целевой адрес:
  - В ou=lists (атрибут rfc822MailMember): обрабатывает как рассылку – в т.ч. передаётся в Cyrus в shared folder
  - В ou=aliases (атрибут rfc822MailMember): раскрывает как почтовый алиас
  - В ou=users (атрибут mailLocalAddress): передаёт в Cyrus в соотв. папку

LDAP + IMAP

- Аутентификация пользователей – через LDAP
- Специальный Perl-backend (в отдельном consumer-сервере) ловит всё изменения в дереве и отражает их в Cyrus
  - slapd-perl(5)
  - Cyrus::IMAP::Admin
- При добавлении пользователя или при добавлении атрибута mail, содержащего домен, который находится в dc=XXX,ou=mail – в Cyrus создаётся соответствующий почтовый ящик.

LDAP + IMAP: рассылки

- При добавлении рассылки в ou=lists,dc=XXX,ou=mail создаётся соотв. shared folder в Cyrus
- При изменении прав на рассылку (добавление/удаление пользователей из дочерних объектов cn=read, cn=write) – соответственно обновляются ACL на shared folder’е

Учёт сетей

- Сети – их много!
  - Много внутренних (`10.0.0.0/8`)
  - Немало внешних (выделенных в RIPE)
- Их надо учитывать, хранить в едином месте:
  - Основные параметры (адрес и маску)
  - RIPE-данные
  - VLAN, routing, etc.

Сети: DIT

```
o=company
   ou=networks
      cn=10.0.0.0/8
         cn=10.99.0.0/16
      cn=88.55.66.0-88.55.66.255
      ...
```

Сети: пример

```
dn:
   cn=10.99.0.0/16,...,ou=networks,o=company

objectClass: ipNetwork, intraNetConfig
ipNetworkNumber: 10.99.0.0
ipNetmaskNumber: 255.255.0.0
vlanNumber: 555
defaultRoute: 10.99.0.1
owner: uid=vpupkin,ou=users,o=company
```

Сайты (sites)

- Сайт – это scope of administration
- Содержит ресурсы, объединённые по логическому (corp, devel, infra) или проектному (hosting, spylog) принципу
- Ресурсы:
  - хосты, группы доступа, настройки для приложений

Сайты: DIT

```
o=company
   ou=sites
      ou=foobar
      ou=hosting
      ou=infra
      ou=spylog
      ...
```

Типичный сайт: DIT

```
o=company,ou=sites,ou=foobar
    ou=hosts
    ou=groups
     ou=shellusers
     ou=web
    ou=virtual
    ou=sudoers
```

Хосты

- Хостов ещё больше чем сетей
  - Их часто просто теряют
  - Выделяют повторно одни и те же ip-адреса
  - Не всегда ясна связь с реальным железом, со складским учётом и бухгалтерией
  - Инвентаризация – нужна!
  - Хочется хранить доп. параметры (carp vhid)

Сайты: хосты

```
o=company,ou=sites,ou=foobar
    ou=hosts
     ou=jails
          cn=www01.int.foobar.ru
     ou=mgmt
          cn=x0666.mgmt
     ou=carp
```

Хост: mainhost

```
dn: cn=x0666,ou=mgmt,ou=hosts,
  ou=foobar,ou=sites,o=company

objectClass: device,ieee802Device,ipHost,intraHost
cn: x0666.mgmt
ipHostNumber: 10.99.10.4
macAddress: 00:35:1A:15:17:42
hostType: main
datacenterServerID: srv_01018
l: ДЦ Алтуфьево
```

Хост: jailhost

```
dn: cn=www01.int.foobar.ru,ou=jails,ou=hosts,
  ou=foobar,ou=sites,o=company

objectClass: ipHost, intraHost
cn: www01.int.foobar.ru
ipHostNumber: 10.99.20.2
hostType: jail
owner:
  cn=x0666,ou=mgmt,ou=hosts,ou=foobar,ou=sites,o=c
  ompany
manager: uid=vpupkin,ou=users,o=company
```

Хосты и DNS

- Хосты заводятся в LDAP одной командой

```
# ldap_ctl --create --host XXX ...
```

- Далее – перегенерятся зоны:

```
# dns_ctl
  --process foobar.zone
  --process 10.99.10.0-24
```

- Думаем (!), коммитим в SVN, выкатываем обновления на authorative NS servers

Хосты и DNS

- Если откуда-то надо срочно понять что это за хост:

```
# host -t TXT www01.int.foobar.ru

www01.int.foobar.ru descriptive text
  "manager:
  uid=vpupkin,ou=users,o=company“
www01.int.foobar.ru descriptive text
  "mainhost: cn=x0666,ou=mgmt,ou=hosts,
  ou=foobar,ou=sites,o=company"
```

Группы доступа

- Группы доступа - самый популярных способ в реализации авторизации
- Контроль доступа к:
  - Хостам (по SSH)
  - Веб-приложениям (например, всё Apache-based)
- Используем класс groupOfUniqueNames

Сайты: хосты

```
o=company,ou=sites,ou=foobar
    ou=groups
      ou=shellusers
            cn=mgmt-dev
      ou=web
            cn=bugzilla
```

Группа: пример

```
dn: cn=mgmt-dev,ou=shellusers,ou=groups,
  ou=foobar,ou=sites,o=company

objectClass: groupOfUniqueNames
cn: mgmt-dev
uniqueMember:
  uid=vpupkin,ou=users,o=company
description: Доступ на dev mainhosts
```

PAM

/etc/pam.d/sshd , /etc/pam.d/system

```
# auth
auth    sufficient  /usr/local/lib/pam_ldap.so
            no_warn try_first_pass
auth    required pam_unix.so
            no_warn try_first_pass
# account
account required /usr/local/lib/pam_ldap.so
      ignore_authinfo_unavail ignore_unknown_user
account required pam_unix.so
```

/etc/nsswitch.conf

```
passwd: cache files ldap
```

nss_ldap

/usr/local/etc/nss_ldap.conf

```
base   ou=users,o=company
uri ldap://ldap.company.ru
binddn uid=pam,ou=virtual,ou=foobar,ou=sites,o=company
bindpw 26a9e1b8df74606eaafa2dde8f8964c1

pam_login_attribute uid
pam_member_attribute uniqueMember

pam_groupdn   cn=mgmt-dev,ou=shellusers,ou=groups,
   ou=foobar,ou=sites,o=company

sudoers_base ou=sudoers,ou=foobar,ou=sites,o=company
```

OpenSSH

/usr/local/etc/ssh/sshd_config

```
UseLPK      yes
LpkLdapConf   /usr/local/etc/nss_ldap.conf
LpkServers    ldap://ldap.company.ru
LpkForceTLS   yes
LpkUserDN     ou=users,o=company

LpkBindDN
  uid=pam,ou=virtual,ou=foobar,ou=sites,o=compan
  y
LpkBindPw     26a9e1b8df74606eaafa2dde8f8964c1
```

Sudo + LDAP

- Sudo – стандарт де-факто для контроля выдачи привилегий (в т.ч. под root'ом)
- Настройки файла sudoers можно хранить в LDAP
  - Почти без потери функциональности
  - При этом для надёжности локальный файл sudoers можно просто запретить читать

Сайты: sudoers

```
o=company,ou=sites,ou=foobar
    ou=sudoers
      cn=defaults
      cn=root@mgmt-dev
      cn=%www@dev-web
```

Sudo: defaults

```
dn: cn=defaults,ou=sudoers,
  ou=foobar,ou=sites,o=company

objectClass: sudoRole
cn: defaults
sudoOption: !env_reset
sudoOption: ignore_local_sudoers
```

Sudo: пример

```
dn: cn=root@mgmt-dev,ou=sudoers,
  ou=foobar,ou=sites,o=company

objectClass: sudoRole
cn: root@mgmt-dev
sudoCommand: ALL
sudoHost: x0666.mgmt
sudoOption: !authenticate
sudoRunAs: root
sudoUser: vpupkin
```

Виртуальные пользователи

- В LDAP пользователь – это тот, кто:
  - Есть как объект в DIT
  - Имеет пароль в атрибуте userPassword
- Как следствие он:
  - Может делать bind к DIT
  - Т.е. получает доступ к дереву с некими правами

Сайты: вирт. users

```
o=company,ou=sites,ou=foobar
    ou=virtual
      uid=inventory-bot
      uid=pam
      uid=wiki-bind
      uid=jabber-notifier
```

Вирт. user: пример

```
dn: uid=pam,ou=virtual,
  ou=foobar,ou=sites,o=company

objectClass: account, simpleSecurityObject
uid: pam
userPassword:
  {SSHA}TgTWBI+nMOOYC8OMLIZaHufQjFDsS2UHzbx12Q==
description: PAM binding
```

Apache + LDAP: Subversion

```
AuthBasicProvider ldap
AuthType          Basic
AuthName          "Subversion"
AuthLDAPBindDN uid=apache-svn,ou=virtual,ou=devel,ou=sites,o=company
AuthLDAPBindPassword    09a5f58cdd4fbd038d86703d0984d604
AuthLDAPURL    ldaps://ldap.company.ru/ou=users,o=company?uid?sub

AuthLDAPGroupAttribute    uniqueMember
AuthLDAPGroupAttributeIsDN    on
Require ldap-group
        cn=subversion,ou=groups,ou=devel,ou=sites,o=company
AuthLDAPRemoteUserAttribute  uid
AuthzLDAPAuthoritative    on

AuthzSVNAccessFile/usr/local/etc/subversion/access.conf
```

OpenLDAP: подсистемы

- OpenLDAP Software 2.4 Administrator's Guide
- Backend - hdb
- Overlays:
  - accesslog – протоколирование изменений (в отдельное поддерево)
  - unique – гарантия уникальности объекта: uid, uidNumber, mail, mailLocalAddress, ipHostNumber
  - memberof – автоматические backlink ссылки

OpenLDAP: ACL

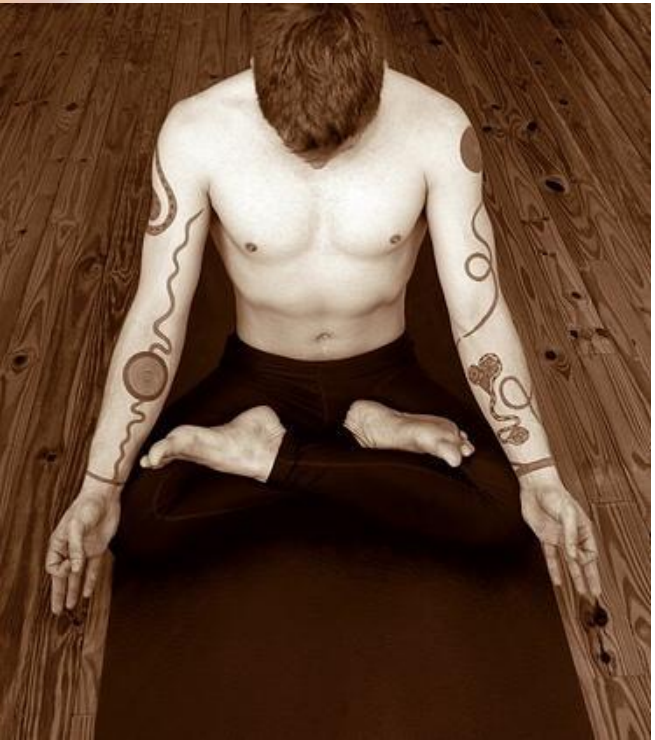
- Rule-based Access Control Lists - очень мощный синтаксис
  - «Access Control», slapd.access(5)
- Можно делать тесты
  - slapacl(8)
- Есть экспериментальные ACI - контроль доступа на уровне объекта(ов), когда ACL хранятся прямо в дереве. Не рекомендую.

OpenLDAP: репликация

- Replication:
  - LDAP Sync Replication
  - Delta-syncrepl replication
  - N-Way Multi-Master replication
  - MirrorMode replication
  - Syncrepl Proxy
- Мы используем MirrorMode
  - Есть куда расти

Инструментарий

- Command line forever!
  - ldapsearchldapsearch, ldapaddldapsearch, ldapadd, ldapmodify, ldapdelete, …
  - ldapvi – вся мощь Vim для LDAP!
- Apache Directory Studio – Java GUI
- Внутренние разработки:
  - ldap_ctl, perl backends, …



III. Размышления

Где использовать LDAP ?

- Все реализации LDAP оптимизированы на массовые чтения (в отличии от SQL)
  - Dynamic Directory Services – это извращение
- Хорошая репликация
- Время «запрос-ответ» можно уменьшать
- Это служба каталогов, а не lookup сервис (в отличии от DNS)
  - Т.е. удобство схем, мощный поиск

Почему не SQL ?

- SQL – это реляционная модель
- out-of-box:
  - Много полезных и de-facto стандартных схем
  - Аудит, репликация
- Стандартный протокол – много software и hardware его умеют
- Это служба каталогов, а не lookup сервис (в отличии от DNS)
  - Т.е. удобство схем, мощный поиск

Где мы не используем LDAP

- Хранение в LDAP полностью зон DNS
  - Есть схемы: DNSZone, DNSDomain2
  - Поддерживается в BIND, PowerDNS
  - Очень неудобно править *YMMV
- Интеграция с DHCP
  - Пока не было нужно, возможно будет для PXE

Что в планах

- Хранение в LDAP корпоративных PGP Keys
  - gnupg 2.x так умеет
- Реализация шаблонов (templates) для заведения учётной записи / её upgrad'а
- PKI – хранить внутренние SSL-сертификаты

Вопросы?

(по теме доклада)

http://protey.ru/blog/2009/04/rootconf-2009.html

Сергей Скворцов
skv@protey.ru