Fedora Directory Server

Kerberos Single Sign-On

Pitfalls

v. 29/03/2009

Содержание

Предисловие

В этом документе содержится информация по организации совместной работы Fedora Directory Server 1.1.0, MIT Kerberos 1.6.0 и системы Single Sign-On на этой основе. Система дает возможность использовать один пароль для доступа к различным службам, таким как почта,

- веб-страницам с авторизацией (tiki-wiki),
- cvs, svn,
- samba shares,
- прозрачной авторизацией на http proxy server squid,
- а также авторизация на рабочих станциях под управлением Linux (в потенциале и с другими POSIX-совместимыми OS) и Windows.

Корпоративную адресная книга позволяет пользователям искать информацию о сотрудниках, их email'ах, телефонах и т. д. в единой базе. У администраторов появляется возможность централизованно задавать пароли, включать пользователей в группы, выдавать права на исполнение sudo. Данная работа была проведена в рамках "легализации" программного обеспечения в фирме. Настоящий документ не претендует ни на исчерпывающую полноту, ни универсальность, ни на совершенство реализации, однако, надеюсь, может быть полезен тем, кто настраивает что-то подобное. Конструктивные замечания и усовершенствования приветствуются. Связаться со мной можно по

jabber: crypt [at] jabber.org

Необходимые знания

Предисловие можно образно назвать ложкой мёда, и собственно мёд на этом заканчивается. Данный документ нельзя рассматривать как пошаговое руководство для новичков. В нем совершенно отсутствует раздел «Что такое LDAP», не описано то, что описано в каждом втором howto и т.д. Это скорее описание граблей, на которые наступил автор. Зато документ выгодно отличается от поверхностных обзоров Fedora Directory Server, пример которых можно найти, скажем, в «Системный администратор» №1, январь 2008. Будут упомянуты те сложности, которые возникли и способы их решения. Документ рассчитан на читателя, который комфортно чувствует себя в UNIX системах, знает, что такое LDAP, Kerberos, PAM модули и Google. Базовыми источниками являются документация на следующих сайтах:

• http://www.openldap.org/software/

• http://web.mit.edu/Kerberos/

• документ с названием Red Hat Directory Server Administration Guide

```
вот так в тексте помечена справочная информация
```

```
вот так обозначаются команды и текст в терминале
```

Об использованном ПО

Fedora Directory Server (FDS) представляет собой купленный Red Hat в 1996 г. Netscape Directory Server (NDS). Более подробно о этом написано, например, в Wikipedia. OpenLDAP ответвился от NDS, поэтому все три очень похожи. К FDS идёт GUI на Java. Очень медленный, но в целом подходит, чтобы делать единичные изменения и обозревать картину в целом. Изначально для реализации Single Sing-On (SSO) рассматривался также OpenLDAP, но в конечном итоге выбор был сделан в пользу FDS, поскольку у Red Hat была заявлена работа с MIT Kerberos, совместная работа с Microsoft AD (в данном документе не рассматривается) и предполагалось, что частично работа по интеграции уже проделана. Кроме того, в организации на серверах использовался Fedora Linux. Однако при должных навыках все описанное ниже может быть проделано и с OpenLDAP.

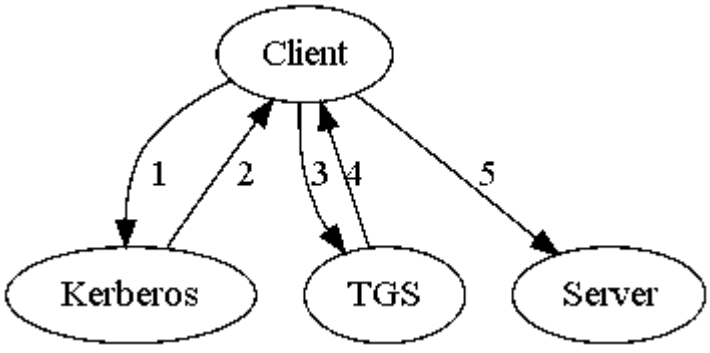
FDS проще всего установить в систему от RedHat. Установка на другие дистрибутивы предполагает дополнительные сложности в виде исправления путей, разрешения зависимостей, сложность обновления и т.д. Вероятно в таком случае лучше использовать OpenLDAP. В нашем случае в качестве дистрибутива на сервере, отведённом под DS, был использован Fedora Core. Все необходимое ПО (кроме некоторых ldap schema) присутствует в репозитарии.

Данный документ написан в OpenOffice, схемы сделаны при помощи dot из пакета graphviz (http://www.graphviz.org/)

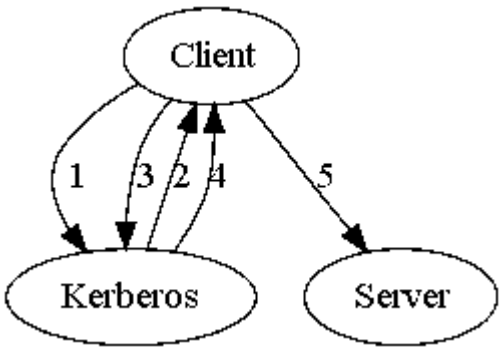
О том, как работает Kerberos

На всякий случай, кратко о работе протокола Kerberos. Если Client желает связаться с Server, то

1) он связывается со службой Kerberos для получения временного билета, тикета (ticket. Правильное русское название - мандат) для доступа к Службе выделения мандатов (Ticket-Granting Service, TGS).
2) Ticket высылается в зашифрованном виде
3) client запрашивает у TGS возможность соединиться с Server
4) Если все в порядке, TGS отсылает ticket на соединение
5) Далее Client предъявляет этот ticket и свой идентификатор Server'у



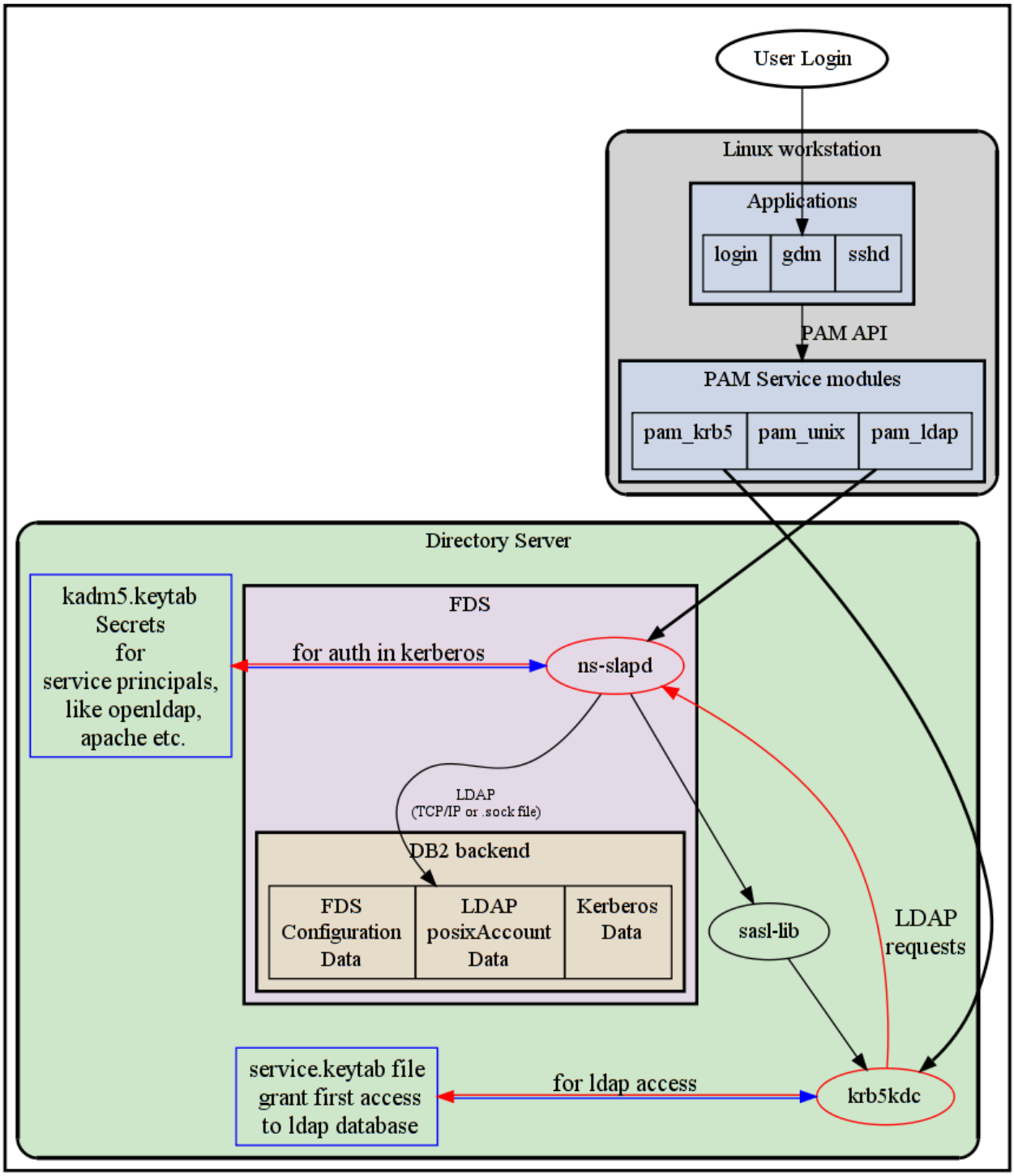
Поскольку в MIT Kerberos сервер Kerberos и TGS будут в одном демоне krb5kdc, то схему можно упростить до такой.



Так что при просмотре вывода klist на стороне клиента мы будем видеть два тикета.

О проблеме курицы и яйца

При настройке Directory Server'a возникает следующее затруднение. FDS может разрешать доступ, консультируясь с базой Kerberos. Kerberos может хранить свои данные в базе LDAP, составной части FDS. В случае если к FDS поступит запрос к данным LDAP каталога, он обратится к демону Kerberos (KD), чтобы выяснить есть ли права на доступ, а Kerberos в свою очередь запросит у FDS данные из LDAP. Решении состоит в том, что FDS будет использовать так называемый keytab file, на основании которого KD разрешит доступ. А для KD в свою очередь будет создана запись в LDAP и специальный stash-file. Фактически в обоих файлах записаны пароли. Для пояснения картины в целом прилагается схема.



Данные о паролях пользователей могут храниться, как в LDAP (userPassword), так и в базе Kerberos. Некоторое ПО не имеет поддержки Kerberos, но может LDAP+SSL/TLS. Так что все ПО, которое может использовать выгоды SSO, можно разделить по способу, как оно будет аутентифицироваться:

- LDAP, simple bind, plain text, небезопасно, подходит только для анонимного подключения.
- LDAP, SSL/TLS, не совместимо с реализацией Kerberos.
- Kerberos, хеши и информация протокола на более низком уровне не шифруется,

поэтому тоже можно считать plain text, хотя критических данных не передается, поэтому можно считать безопасным.

- Резервный, но надёжный вариант: можно использовать системную авторизацию через PAM. Но тикеты Kerberos не используются, пользователям требуется вводить пароль вручную.

Поддержка того или иного способа аутентификации может быть встроена в сам клиент или реализована через тот или иной уровень абстракции. Следующие понятия тесно связаны с темой документа.

SASL(Simple Authentiction and Security Layer) - слой (protocol framework) между OpenLDAP (FDS) и Kerberos для безопасной аутентификации LDAP-сервера в Kerberos. SASL использует kerberos через плагин, который использует GSSAPI.
GSSAPI (Generic Security Services Application Programming Interface). Если SASL - это уровень сетевого протокола, то GSSAPI - API. Клиент и сервер вначале используют SASL, чтобы договориться, что используют GSSAPI (а не plain text, например), потом GSSAPI plugin на обеих машинах используется, чтобы обмениваться по Kerberos протоколу.
SSPNEGO Поскольку Microsoft использует NTLM для SSO, они изобрели механизм, аналогичный GSSAPI механизм для согласования протокола аутентификации. К счастью MIT Kerberos поддерживает SPNEGO с версии 1.5 благодаря реализации, пожертвованной SUN. Поэтому можно использовать библиотеку GSSAPI на стороне сервера (касается только Linux).
NSS (Name Service Switch) используется, чтобы определить, откуда будет браться информация об именах: из локальных файлов, DNS, NIS, LDAP.
Principal - термин для обозначение объекта в базе Kerberos.
Distinguished Name (dn) - термин для обозначения объекта в базе LDAP.
userPassword — поле объекта LDAP, используется при simple bind с LDAP (возможно использование SSL/TLS), при SASL binds using Digest-MD5/CRAM-MD5, а при SASL/GSSAPI and SASL/EXTERNAL не используется.

Если ПО имеет поддержку GSSAPI, то все просто. CVS, SVN (с версии 1.5), Apache через соответствующий модуль и множество других программ. В случае же tiki-wiki, Lotus Domino поддержки нет и приходится обращаться напрямую к каталогу LDAP. Поэтому возникает вопрос о синхронизации информации о паролях, которая храниться в LDAP (**userPassword)** и в базе KD.

Вопрос является актуальным и плохо документированным до сих пор. В частности в случае с OpenLDAP, говорят, можно указывать userPassword в виде {KERBEROS}principal@REALM и это сработает с pam_krb5 модулем. Дискуссия на эту тему в этой рассылке openldap:

http://www.openldap.org/lists/openldap-software/200502/msg00470.html

Здесь http://www.openldap.org/faq/data/cache/944.html написано, что

{KERBEROS} заменено на {SASL}. Если кто-то знает, работает ли это и с каким ПО, я был бы рад узнать.

Поскольку не было гарантировано, что хак подобного рода будет работать в любом случае, я решил просто написать скрипт, который при создании пользователя будет задавать один пароль в обоих местах. При настройке службы каталога важным является предварительное планирование. После создания объектов в LDAP уже невозможно изменить их базовый objectClass и состав основных свойств-полей, поэтому может случиться, что если вы по ходу настройки решите использовать LDAP для хранения дополнительного рода информации, это можно будет сделать только путём дампа старых данных, их модификации и заливки

обратно. Ценный документ на тему планирования называется gg://"LDAP Schema Design by Andrew Findlay". В числе советов: не модифицировать стандартные схемы, создавать объекты на основе structural класса, а потом расширят за счёт добавления auxilary.

План установки FDS был следующий:

1. ставился FDS,
2. создавался временный пользователь для работ,
3. подключались дополнительные ldap схемы,
4. сразу после этого его LDAP каталоге создавалась база Kerberos,
5. решались вопросы с доступом FDS к KD и наоборот
6. вносились данные о реальных пользователях
7. устанавливались и проверялись LDAP ACL (здесь не описывается)

Об установке FDS.

В случае с дистрибутивом от Red Hat происходит легко, быстро и просто. При установке можно согласится на инсталяцию записей-примеров. Согласно потребностям создаётся общая структура LDAP каталога. В моем случае это People, Groups и Special Users для сотрудников, групп, и учётных записей, которые должны иметь большие полномочия, и принадлежат службам. kdc-service - под этой записью KD будет считывать данные из LDAP. adm-service - в документации к openLDAP предлагается создать, но у меня она не используется.

```
# Special Users, example.com
dn: ou=Special Users,dc=example, dc=com
objectClass: top
objectClass: organizationalUnit
ou: Special Users
description: Special Administrative Accounts
# kdc-service, Special Users, example.com dn: uid=kdc-service,ou=Special Users,dc=example, dc=com
givenName: kdc-service
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
sn: this is for kerberos connect to ldap
cn: kdc-service this is for kerberos connect to ldap
uid: kdc-service
userPassword:: e1NTSEF9TFoxTlBIRGQvemg1WU9yaHhnb1BBBBBBBBBBSSXBqTZVOFE9PQ==
# adm-service, Special Users, example.com
dn: uid=adm-service, ou=Special Users,dc=example, dc=com
uid: adm-service
givenName: adm-service
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
sn: this is for kerberos
cn: adm-service
userPassword:: e1NTSEF9TAAAAAAAZitYT3ppOHZabXl4OGtOV1orNXh5R0loVi9NzIrVEE9PQ==
```

Добавлялась schema, которая необходима, чтобы хранить данные Kerberos. Таких схем в интернете можно найти две разновидности. kdb5_ldap_util использует схему от Novell, часто идущую под названием kerberos.ldif, основной objectClass - krbPrincipal. Также можно столкнуться со схемой, которая используется в Heimdal, часто называется krb5-kdc.schema и имеет основной objectClass krb5Principal.

```
kdb5_ldap_util allows an administrator to manage realms, Kerberos services and
ticket policies.
```

О схемах LDAP в FDS

К этому документу я прилагаю использованные схемы. Те, которых нет в пакете FDS, я брал в пакете openldap. Особенно хороша поставка от Mandrake Linux, там набор схем больше, чем, скажем, в Fedora. Хотя схемы FDS отличаются от схема OpenLDAP, но несильно и их можно сконвертировать скриптом ol-schema-migrate.pl. Скрипт есть в Приложении и в интернете. Хотя об этом и не везде сказано, но чтобы сервер FDS запустился, помогала удаление всех комментариев из файла схем.

Об установке MIT Kerberos

Создаем realm kerberos. Параметр '-H' может иметь в качестве значения 'ldap:///' - simple bind, 'ldaps:///' - ssl/tls, 'ldapi:///' - связь через сокет. '%2f' заменяет слеш. FDS по какой-то причине сокет у меня не создал. Так что это касается openLDAP. Здесь и далее настоящие названия DNS домена и Kerberos realm заменены на example.com и EXAMPLE.COM.

```
kdb5_ldap_util -D "cn=crypt,ou=People,dc=example,dc=com" \
-H "ldap:///" create -r "EXAMPLE.COM"
```

Напишем конфигурационные файлы (есть в приложении):

- /etc/krb5.conf - Configuration file for Kerberos systems (clients and servers)
- /etc/kdc.conf - Configuration file for Kerberos servers
- /etc/krb5kdc/kadm5.acl (/var/kerberos/krb5kdc/kadm5.acl) - Access Control List file

Создаем stash-file, чтобы KD мог получить доступ до LDAP:

```
kdb5_ldap_util -D "cn=administrator,ou=People,dc=example,dc=com" stashsrvpw \
-f /var/kerberos/krb5kdc/service.keyfile cn=kdc-service,dc=Special Users,dc=example,dc=com \
-H ldap:///
```

После этого можно запускать KD.

Возможные сообщения об ошибках MIT Kerberos

```
Server error - while initializing database for realm EXAMPLE.COM
```

KD не может соединиться со службой LDAP. Может быть неправильно указан адрес LDAP сервера. Сокет не тот или еще что.

```
Unable to access Kerberos database - while initializing database for realm EXAMPLE.COM
```

KD не может получить доступ к данным внутри базы. Тут как раз важно проверить при помощи ldapsearch, что kdc-service получает доступ и ACL не мешают.

Теперь сделаем, чтобы FDS мог получить доступ в базу Kerberos при помощи keytab файла. Используем kadmin.local.

kadmin и kadmin.local предоставляют command-line интерфейс для администрирования Kerberos V5. Они абсолютно идентичны, но kadmin.local не требует авторизации. Можно использовать либо интерактивно, либо в скриптах: kadmin.local -q command.

```
addprinc -randkey ldap/192.168.1.6@EXAMPLE.COM
```

Добавляет принципала в базу Kerberos. В записи может фигурировать, как ip адрес, так и DNS имя. В данной команде ldap - это часть имени, соответствует типа сервиса (*ServiceClass*, **SPN, Service Principal Name — в терминах MS**), задается по смыслу. Например, для apacha и squid'a была бы http, для NFS - nfs, DNS - dns и т.д. -randkey для генерации произвольной последовательности вместо ввода пароля вручную. Следующей командой мы экспортируем эту последовательность в файл и даже не обязательно её знать. При работе я заметил, если пароль задавать вручную, а потом выполнять ktadd, пароль меняется. Так что при добавлении

в keytab файл всегда используется случайная последовательность.

```
ktadd -k /var/kerberos/krb5kdc/kadm5.keytab ldap/192.168.1.6@EXAMPLE.COM
```

kadm5.keytab также может просто лежать в /etc. Важно, чтобы во всех конфигурационных файлах использовался нужный файл, а не то, что по умолчанию.

Добавим политику паролей по умолчанию:

```
addpol -maxlife "36 month" -minlength 8 -minclasses 2 default
```

Смена пароля будет требоваться каждые 36 месяцев, minclasses означает необходимость использовать хотя бы два различных набора символов, т.е. хотя бы цифры и буквы.

Если собираетесь администрировать удаленно при помощи kadmin, то потребуется создать административного пользователя и написать файл kadm5.acl такого вида.

```
*/admin@example.com    *
```

В данном случае даются все права (*) для admin@example.com.

Проверка

Чтобы просто проверить, как работает доступ к LDAP, можно выполнить ldapsearch с параметрами для simple bind: -x -w или -W.

```
ldapsearch -vv -D "uid=crypt,ou=People, dc=example,dc=com" -xwSecretPasswrdForCrypt -H \
"ldap://192.168.1.6" -b "ou=People, dc=example,dc=com"
```

Чтобы проверить, работу KD, используйте kinit, чтобы получить тикет, а затем ldapsearch -Y GSSAPI, имя пользователя при этом указывать уже не надо.

```
kinit crypt
ldapsearch -vv -Y GSSAPI  -H "ldap://192.168.1.6" -b "ou=People, dc=example, dc=com"
```

Возможные ошибки при запуске ldapsearch

```
ldap_sasl_interactive_bind_s: Invalid credentials (49) additional info: SASL(-1): generic failure:
GSSAPI Error: Unspecified GSS failure. Minor code may provide more information (No principal in
keytab matches desired name)
```

Необходимо проверить, читается ли kadm.keytab.

```
-rw-r----- 1 root ldap 1890 2008-04-30 18:54 /var/kerberos/krb5kdc/kadm5.keytab
```

В логах FDS может быть:

```
sasl(2): GSSAPI Error: Unspecified GSS failure.  Minor code may provide more information (Wrong
principal in request) => send_ldap_result 49::SASL(-13): authentication failure: GSSAPI Failure:
gss_accept_sec_context
```

и при этом в логах KD:

```
ds..office.example.com krb5kdc[12023](info): TGS_REQ (2 etypes {1 3}) 192.168.1.154: ISSUE:
authtime 1207572455, etypes {rep=1 tkt=16 ses=1}, crypt@EXAMPLE.COM for
ldap/ds.office.example.com@EXAMPLE.COM
```

Это означает, что согласно принципу работы Kerberos протокола, пользователь crypt успешно запросил тикет на получение тикета для доступа к службе ldap/ds.office.example.com, успешно получил этот тикет и передал его FDS. А когда FDS, используя библиотеку cyrus-sasl, извлекает из него данные, необходимые, что аутентифицироваться в KD, то оказывается, что они, так называемый security context (gss_accept_sec_context), неверны. Коды ошибок даны в приложении.

Наиболее часто при такого рода ошибках следует проверить синхронизацию времени на сервере и клиенте или имена DNS. Если с синхронизацией по NTP все ясно, то с резолвингом имен следует помнить, что имя, которое клиент присылает в тикете и имя, под которым видит себя KD, должны совпадать, альясы DNS (записи типа CNAME) не допускаются, не забывайте про /etc/hosts. Сложность может возникнуть, если у вас для одного хоста назначено несколько имен.

В стандарте DNS допускаются специальные SRV записи для службы Kerberos.

```
_kerberos._udp          SRV       0 0 88 ds
_kerberos-master._udp   SRV       0 0 88 ds
_kerberos-adm._tcp      SRV       0 0 749 ds
_kpasswd._udp           SRV       0 0 464 ds
```

Настройка аутентификации и авторизации на пользовательских станциях: Linux

Делается при помощи PAM. По поводу этой подсистемы масса информации, мне более всего понравилась книга Pluggable Authentication Modules, Kenneth Geisshirt. При этом аутентификация выполняется при помощи Kerberos pam модуля. А данные о пользовательской учётной записи запрашиваются из LDAP при помощи модуля NSS, nss_ldap. PAM LDAP, который мог бы использоваться для аутентификации через LDAP не используется. В этой схеме есть один недостаток. Для работы NSS LDAP требуется пользователь-посредник, который получает доступ в LDAP при помощи keytab файла и считывает данные об учётной записи. При этом у этого пользователя должны быть права на чтения **всего** дерева Kerberos.

! Потенциально получить дамп базы kerberos может любой привилегированный пользователь на локальной машине от имени пользователя-посредника.

В базе **содержатся хеши паролей пользователей.**

На мой взгляд было бы более резонно аутентифицировать пользователя через PAM Kerberos, потом отослать его тикет-кеш LDAP серверу, который бы извлек оттуда имя пользователя и проверил его наличие в базе LDAP. Т.е. использовать ту же схему, что действует, когда пользователь уже вошел. Если кто-то знает, почему для NSS LDAP обязателен пользователь-посредник, пусть сообщит мне об этом.

Файл /etc/pam.d/system-auth является базовым и часто включается в файлы других служб или на него просто делается симлинк. Например:

```
root@orb ~ # cat /etc/pam.d/sshd
#%PAM-1.0
auth       include      system-auth
account    required     pam_nologin.so
account    include      system-auth
password   include      system-auth
session    optional     pam_keyinit.so force revoke
session    include      system-auth
session    required     pam_loginuid.so
```

Собственно, в него я и внёс изменения, его вы можете найти в Приложении. Здесь я включу комментарии.

```
#%PAM-1.0
auth        required      pam_env.so # устанавливает параметры среды
auth        sufficient    pam_unix.so nullok try_first_pass # пароль может быть пустым, попытаться
использовать тот пароль, который был передан предыдущему модулю
auth        sufficient    pam_krb5.so use_first_pass forwardable # использовать предыдущий пароль,
поставить kerberos flag разрешающий форвардинг тикета. поставил на всякий случай. при настройке
бывает полезно использовать опцию debug. и не только здесь.
auth        required      pam_deny.so # nocomments

account     required      pam_access.so
account     [default=bad success=ok user_unknown=ignore service_err=ignore system_err=ignore]
pam_krb5.so # если проверка провалилась тут — ничего страшного. проверяется не истек ли пароль,
временные ограничения и т.п.
account     required      pam_access.so


password    requisite     pam_cracklib.so try_first_pass retry=3
password    sufficient    pam_unix.so md5 shadow nullok try_first_pass use_authtok # нужен для
смены пароля пользователем через kpasswd.
password    sufficient    pam_krb5.so use_authtok
password    required      pam_deny.so

session    required    pam_mkhomedir.so skel=/etc/skel/ umask=0022 # создает home dir, если еще не
существует. К сожалению, в не позволяет динамически формировать содержимое. Но при желании этот
функционал несложно дописать.
session     optional      pam_keyinit.so revoke
```

```
session     required      pam_limits.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session     required      pam_unix.so
session     optional      pam_krb5.so # смысловой нагрузки не несет.
```

Эта конфигурация позволяет использовать как /etc/passwd, так и Kerberos.

Настройка NSS LDAP осуществляется в /etc/ldap.conf. Есть в приложении, здесь с комментариями.

```
base dc=example, dc=com
uri ldap://ds.office.example.com
# как раз тот самый посредник, который имеет полный доступ.
binddn uid=proxyservice, ou=People, dc=example, dc=com
timelimit 120
bind_timelimit 120
idle_timelimit 3600
pam_check_host_attr no # если подключить схему ldapns (идет вместе с pam ldap), то можно добавить
специальный host атрибут и таким образом ограничить доступ определенного пользователя к
определенному хосту. однако в нашем случае это не работает, т.к. мы используем для аутентификации
ldap kerberos. использован другой метод, об этом см. ниже.
nss_base_passwd ou=People,
nss_base_shadow ou=People,
nss_base_group ou=Groups, # группы должны содержать objectClass: posixGroup
sudoers_base ou=People,
nss_initgroups_ignoreusers
root,ldap,named,avahi,haldaemon,dbus,radvd,tomcat,radiusd,news,mailman,nscd # можно было бы
использовать больше от NIS, но в данном случае роли эта строка не играет
#ssl start_tls # если бы использовался просто pam ldap, то был бы смысл защищать канал. В нашем
случае Kerberos протокол не работает поверх SSL/TLS.
ssl off
tls_cacertdir /etc/openldap/cacerts
pam_password md5
tls_checkpeer no # если сложно с установкой сгенерированных на сервере сертификатов, то можно
выключить
use_sasl on # тут мы включаем работу через Kerberos
sasl_auth_id proxyservice@example.com
pam_sasl_mech GSSAPI
#pam_check_host_attr yes
pam_filter &(objectclass=posixAccount)(host=ares) # к сожалению, в нашем случае эта проверка не
срабатывает, атребут не проверяется.
#pam_check_service_attr yes
bind_policy soft # в случае, если LDAP сервер недоступен, можно легко залочить машину, если не
включить это
#sudoers_debug 1 # для отладки работы sudo через ldap
#krb5_ccname FILE: /tmp/krb5cc_0 # полезная опция, можно указать cache файл. Но я не
воспользовался.
```

И последний, третий файл - /etc/nsswitch.conf. Настраивается очерёдность, как система будет брать информацию о пользователях. Есть в приложении.

proxyservice аутентифицируется в Kerberos при помощи keytab-файла, который создаётся следующим образом.

```
add_principal -randkey proxyservice
ktadd -k /root/proxyservice.keytab proxyservice
```

Потом он переносится на целевую систему. Доступ к нему ограничивается только для root.

Чтобы ticket-cache регулярно обновлялся создаётся запись в cron'e:

```
 */30 * * * *    /usr/kerberos/bin/kinit -k -c /tmp/krb5cc_0 -t /etc/proxyservice.keytab
proxyservice
```

На серверах, где запросы NSS случаются часто (например, почтовый сервер, где постоянно идет спам), количество запросов к DS и соответственно нагрузку на сеть можно снизить, установив демон nscd. С ним следует помнить, что если вы видите ошибку доступа к LDAP или похожую, например

```
nscd: GSSAPI Error: Unspecified GSS failure.  Minor code may provide more information (No
credentials cache found)
```

то следует помнить, что nscd запускается под собственным пользователем и ошибка свидетельствует о проблемах доступа к файлам кеша kerberos. Я просто поправил /etc/nscd.conf и запустил от root.

Кроме того, в случае запуска какого-либо сервиса под отдельным аккаунтом, мы также

должны обеспечить ему доступ к информации о posix account в LDAP. Если этого не сделать, мы столкнемся с ошибкой вида

```
nss_ldap: failed to bind to LDAP server ldap://ds.example.com: Local error
nss_ldap: could not search LDAP server - Server is unavailable
GSSAPI Error: Unspecified GSS failure.  Minor code may provide more information (No credentials
cache found)
```

Так что необходимо либо создавать специальный keytab-файл для такой учётной записи, либо использовать nscd как прослойку, чтобы он обрабатывал запросы об именах.

Для обновления информации в кэше nscd необходимо выполнить при запущенном nscd.

```
nscd -i passwd
```

Иногда, например в случае с apache, данные nscd не используются, учётная запись заблокирована и необходимо каким-то образом все-таки создавать файл с кешем Kerberos. Для этого есть различные возможности.

Замечание о демоне nscd

В процессе эксплуатации выяснилась очень неприятная особенность. Демон nscd падал очень часто. Как временное решение я через крон делал перезапуск и даже в этом случае время от времени у меня были неприятные минуты, когда он не обрабатывал запросы. В конце концов выяснилось, что с этим поведением столкнулся не только я. Денис Власенко, участник проекта busybox, написал альтернативную реализацию демон под названием unscd. Как следует из комментариев к программе, автор пришёл к выводу что nscd, который идет вместе с glibc (в Fedora включили именно эту реализацию. впрочем я не удивлён), изначально имеет задумку, чреватую ошибками. Реализация Дениса компактнее и несомненно стабильнее. Минимально исправив скрипт запуска и конфигурационный файл и все встало на свои места. На данный момент в случае nscd -K, т.е. стандартного способа остановки, демон возвращает ненулевое значение, что может привести к конфузу в скриптах.

Сайт программы: http://busybox.net/~vda/unscd/

Замечание о модуле pam_krb5

Есть несколько реализаций этого модуля. Я бы об этом никогда не узнал, если бы случайно при обновлении с Fedora 8 на Fedora 9 вдруг не отказал вход в систему с использованием этого модуля. Невозможно было получить тикет-кеш при входе по ssh с жалобой на неправильные права в .k5login (об этом файле, см. ниже). Можно было зайти только под локальной учётной записью, а потом сделать kinit. В который раз помянув «корпоративный стандарт», предписывавший использовать Fedora на серверах, потратил много времени на жонглирование опциями модуля, дабы отключить проверку .k5login. Без желаемого результата. Модуль просто перестал работать так, как работал. Как обозначено в man pam_krb5 модуль поддерживается сотрудником Red Hat Nalin Dahyabhai <nalin@redhat.com>.

Тогда я собрал из исходников модуль, который вполне успешно работает, например в Gentoo Linux. Сайт проекта: http://www.eyrie.org/~eagle/software/pam-krb5/

И проблема исчезла.

Замечание о файле .k5login

У обоих реализаций модуля есть полезная особенность. Можно в домашнюю директорию пользователя сложить файл .k5login, куда внести имена всех пользователей, который аутентифицировавшись в Kerberos смогут получить доступ. Это может полезно, как для группы администраторов, чтобы войти как root, так и для других пользователей, где требуется коллективный доступ. Более того, если таких коллективных учётных записей будет достаточно много, то становится очень удобно администрировать (отключать) их из единого центра. При этом запись в .k5login может оставаться и не нужды искать её по всем серверам во всех пользовательских директориях. В той реализации PAM модуля, которую я стал использовать, есть особенность: если файла в домашней директории нет, то проверка не выполняется, но если файл есть, но пустой, то проверка всегда будет завершаться провалом.

Замечание о модуле nss_ldap

```
(SERVICES): nss_ldap: failed to bind to LDAP server ldap://ds.example.com: Local error
(SERVICES)[1728]: GSSAPI Error: Unspecified GSS failure.  Minor code may provide more information
(No credentials cache found)
```

В поиске решения выяснилось следующее.

Выдержка из nssldap.spec файла из nss_ldap-259-3.fc9.src.rpm.

```
%configure \
        --with-ldap=openldap \
        --enable-schema-mapping \
        --enable-rfc2307bis \
        --enable-configurable-krb5-ccname-gssapi
```

В тоже время при сборке модуля есть другие полезные опции:

```
  --enable-configurable-krb5-ccname-env   enable configurable Kerberos V credentials cache name
(putenv method), позволяет задавать имя тикет-кеш файла через переменную среды KRB5CCNAME.
  --enable-configurable-krb5-ccname-gssapi   enable configurable Kerberos V credentials cache name
(gssapi method)
  --enable-configurable-krb5-keytab enable configurable Kerberos V keytab file name, можно задавать
keytab файл.
```

Замечание о модуле pam_mount

Этот модуль можно использовать для автоматического монтирования директорий с файловых серверов samba, NFS и т. п.

sudo

Можно сделать так, что sudo тоже будет использовать LDAP. Правда чувствуется недореализация со стороны разработчиков на тот момент, когда я его смотрел. Устанавливается схема, которая идёт в инсталяции с sudo, сконвертированная для FDS есть в приложении. Пользователям назначаются соответствующие атрибуты, отмечающие есть у них права на sudo или нет. Подробнее см. README.LDAP. Важно! Используется специальный файл /etc/ldap.conf.sudo. Иначе не работает. Поддерживается только очень ограниченное число опция обычного ldap.conf. **Возможен только simple bind**, пароли при передаче не защищены, так что если использовать такую конфигурацию, необходимо создавать очень огранченного пользователя-посредника, с разрешением на чтение только атрибутов sudo.

```
base dc=example, dc=com
ldap version 3
#host ds.office.example.com
#port 389
uri ldap://ds.office.example.com
binddn uid=proxyservice, ou=Special Users, dc=example, dc=com
bindpw secret
sudoers_base ou=People, dc=example, dc=com
ssl off
```

Posix группы

Может быть полезно, когда пользователь аутентифицировался, включить его в локальные группы, типа video, audio, tcpdump и т.д. Для этого в структуре LDAP была создана отдельная ветка, а в ней объекты подобные этому.

```
dn: cn=video, ou=posixGroupsLinux,ou=Groups, dc=example, dc=com
gidNumber: 27
objectClass: top
objectClass: groupofuniquenames
objectClass: posixgroup
uniqueMember: uid=crypt, ou=People, dc=example, dc=com
cn: video
```

Следующая команда печатает ldif-файл с cvs группами. Пользователи отсортированы, дубликаты удалены:

```
grep cvs /etc/group | awk -F: '{ split($4, users, /,/ ); printf("dn: cn=%s, ou=cvs, ou=Groups,
dc=example, dc=com\nobjectClass: top\nobjectClass: groupofuniquenames\nobjectClass:
posixGroup\ngidNumber: %s\n", $1,$3); n=asort(users,userd); for (i=1;i<=n;i++) { if (userd[i] !=
userd[i-1]) printf("uniqueMember: uid=%s, ou=People, dc=example, dc=com\n",userd[i]); } printf("cn:
```

```
%s\n\n", $1); }'
```

split заполняет массив users, asort сортирует и копирует в userd.

Замечание об ограничении доступа пользователей

Если не установить это ограничение, то любой пользователь сможет зайти на любую керберезированную станцию с Linux. Мы не можем использовать `pam_check_service_attr`, см. комментарий к `ldap.conf`. Поэтому будем использовать `access pam module`, вы можете видеть его в `system-auth` в группе `account`. А в `/etc/security/access.conf` добавить следующее:

```
-:ALL EXCEPT root crypt: ALL
```

Это разрешает доступ пользователю `crypt` и `root`.

Более подробно об использовании NIS groups и FDS написано тут:

http://directory.fedoraproject.org/wiki/Howto:Netgroups

Можно, скажем, использовать список доступа на основе групп. К сожалению, в любом случае приходится редактировать конфигурации на клиентской машине.

Настройка аутентификации и авторизации на пользовательских станциях: Microsoft Windows

Windows XP

Я не стал вводить Windows машины в домен Samba при помощи утилиты net, как описано в некоторых руководствах. Возможно, так можно было бы создать rouming profile, но у меня такой цели не было. В моем случае для того, чтобы пользователи могли войти под своей учётной записью, необходимо завести локальных пользователей в Windows, а потом сделать маппинг на них Kerberos аккаунта.

Для этого на сервере kerberos добавляется principal:

```
kadmin.local -q "addprinc -e des-cbc-crc:normal -pw pwd_for_wxp host/cb790.example.com"
```

Важно указать тип хеша des-cbc-crc:normal или des-cbc-md5:normal, которым будут шифроваться данные для этого принципала, иначе вход на windows станцию закончится неуспехом.

А на клиентской windows машине в папку system копируются ksetup и ktpass, которые можно взять из SUPPORT.CAB с инсталяционного диска Windows XP.

```
ksetup используется для конфигурирования realm.
ktpass может устанавливать пароль, account name mappings, and keytab
generation.
```

```
C:> ksetup /setdomain EXAMPLE.COM

C:> ksetup /addkdc EXAMPLE.COM ds.example.com
```

Справка по ksetup выводится по ksetup /?, правда я не обнаружил там тех опций, которые использую и которые указаны в MSDN и тем не менее работают.

Set the local machine account password, as follows:

```
C:> ksetup /setmachpassword pwd_for_wxp

C:> ksetup /mapuser * *
```

Windows Vista

К сожалению, по причине крайней дешевизны и требований некоторого прикладного ПО у нас в сети появилась эта OS. В Windows Vista появилась поддержка AES в реализации Kerberos протокола.

Подробнее:

http://technet2.microsoft.com/WindowsVista/en/library/ba52022e-a5c3-4511-bd7d-4069bb5d3a5e1033.mspx?mfr=true

К счастью утилиты от Windows XP успешно работали, так что процесс настройки аналогичен. Требуется Samba 3.2.0 и выше.

Замечание о Mac OS X

В этой системе есть аналог kinit с GUI - `/System/Library/CoreServices/Kerberos.app`

Вход под учётной записью Kerberos в этой системе я не делал, однако получение тикета для доступа к службам через kinit работает в этой системе обычным образом.

Samba

В Fedora включена версия Samba, которую разработчики предлагают не для промышленного использования, а только для тестирования. Такое положение вещей характерно для данного дистрибутива, но в данном случае на руку, т.к с версии 3.2.0 начинается поддержка аутентификации Vista через Kerberos. Попытка использовать samba из fc8 закончилась core dump'om при попытке соединиться при помощи smbclient -k.

```
!!! Дистрибутив Fedora имеет нестабильный софт!
```

```
-k
          Try to authenticate with kerberos. Only useful in an  Active  Direc-
          tory environment.
```

В сети был поднять Samba PDC непосредственно на сервере DS. PDC выполнял роль сервера паролей для Samba BDC, который и нёс основную нагрузку по передаче файлов.

Создание учётной записи для сервера аналогично предыдущим случаям.

```
addprinc -randkey cifs/cifs.example.com@EXAMPLE.COM
ktadd -k /root/krb5.keytab cifs/cifs.example.com@EXAMPLE.COM
scp /root/krb5.keytab cifs.example.com:/etc/krb5.keytab
```

В LDAP создаётся пользователь для samba с паролем.

```
# samba, Special Users, example.com
dn: uid=samba,ou=Special Users,dc=example, dc=com
uid: samba
givenName: samba
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
sn: samba
cn: samba
userPassword:: e1NTSEF9KWEOV2p5YWF4NmVoMdfEycjhNaUhLYXF4RUdIYmpsWkhVbUhaU0E9PQ=
```

Этот же пароль задается для доступа.

```
 smbpasswd -w secret
```

Использовал следующий конфиг на сервере FDS. Samba в качестве PDC.

Samba PDC config

```
[global]
        netbios name = ds
        realm = EXAMPLE.COM
        workgroup = EXAMPLE.COM
        security = user
        password server = kerberos.example.com
        server string = DS Samba Server
        domain master = yes
        load printers = no
        log file = /var/log/samba/%m.log
        #max log size = 50
        passdb backend = ldapsam:"ldap://localhost"
        socket options = TCP_NODELAY SO_RCVBUF=65536 SO_SNDBUF=65536
        name resolve order = host wins bcast
        dns proxy = yes
        wins proxy = yes
        preferred master = yes
        local master = yes
        hide dot files = No
        domain logons = yes
        template shell = /bin/false
        winbind use default domain = no
```

```
        client schannel = Yes
        server schannel = Yes
        client use spnego = yes
        use kerberos keytab = yes
        idmap backend = ldap:"ldap://localhost"
        ldap admin dn = uid=samba, ou=Special Users, dc=example, dc=com
        ldap group suffix = ou=Groups
        ldapsam:editposix = yes
        ldap suffix = dc=example, dc=com
        ldap user suffix = ou=People
        ldap idmap suffix = ou=winsys, ou=Groups
        ldap machine suffix = ou=Computers
        ldap ssl = off
        # ldap ssl= start tls
        os level = 65
        log level = 10 passdb:5 auth:10 winbind:2
        winbind use default domain = yes
        winbind trusted domains only = yes
        winbind nested groups = yes
        idmap uid = 500-10000000
        idmap gid = 500-10000000
        ntlm auth = no
        lanman auth = no
        client ntlmv2 auth = no
        disable spoolss = yes
        show add printer wizard = no
        case sensitive = no
        default case = lower
        preserve case = yes
[test]
        comment = "test share"
        path = /shared/test
        writeable = yes
        read list = +staff, @BUsers
        write list =
        valid users = @staff, @BUsers
        create mask = 0664
        directory mode = 2775
        public = no
```

Samba BDC config

```
[global]
workgroup = EYELINE
realm = EYELINE.MOBI
netbios name = max
netbios aliases = 01max
server string = Main Samba Server
client schannel = Yes
server schannel = Yes
password server = ds.office.example.com
passdb backend = ldapsam:"ldap://ds.office.example.com"
username level = 3
ntlm auth = No
use kerberos keytab = Yes
log level = 10 passdb:10 auth:10 winbind:10
log file = /var/log/samba/%m.log
name resolve order = host
socket options = TCP_NODELAY SO_RCVBUF=65536 SO_SNDBUF=65536
load printers = No
disable spoolss = Yes
show add printer wizard = No
domain logons = Yes
local master = No
domain master = No
dns proxy = No
wins proxy = Yes
wins support = Yes
ldap admin dn = uid=samba, ou=Special Users, dc=example, dc=com
ldap group suffix = ou=Groups
ldap idmap suffix = ou=winsys, ou=People
ldap suffix = dc=example, dc=com
ldap ssl = no
ldap user suffix = ou=People
idmap backend = ldap:ldap://ds.office.example.com
idmap uid = 500-10000000
idmap gid = 500-10000000
```

```
winbind use default domain = Yes
winbind trusted domains only = Yes
ldapsam:editposix = yes
case sensitive = No
hide dot files = No

[test]
comment = "test share"
path = /shared/test
read list = @staff
read only = No
create mask = 0664
directory mask = 02775
```

Windows to POSIX group mapping

Очень запутанная тема. Здесь с моей стороны могут быть допущены неточности. Основной документ Samba 3 Howto. Основная сложность в том, что у POSIX систем используется в качестве идентификаторов пользователей и групп uid и gid, а в Windows системах свои sid (для пользователей и групп) и rid (для обозначения принадлежности к домену). В итоге, если мы хотим использовать в Samba одну группу, которая будет включать и Windows пользователей и Linux (и Mac), то необходимо, чтобы к POSIX группам был привязан sid, который будет понятен Windows машине. Составлением соответствия занимается служба winbind, которая идет в составе Samba. Есть разные схемы её настройки, об этом написано в Samba 3 Howto. Я использовал в качестве backend'a LDAP, т.к. вижу выгодную сторону в том, что все, что касается FDS хранится в одной базе данных, которую можно удобно реплицировать.

Я в итоге отказался от использования утилиты net для создания мапингов и делал все через ldap объекты.

```
root@orb / # net groupmap list
Administrators (S-1-5-32-544) -> 510
Users (S-1-5-32-545) -> nobody
Domain Admins (S-1-5-21-2391453673-3793323726-2791476427-512) -> 510
Domain Users (S-1-5-21-2391453673-3793323726-2791476427-513) -> 510
Domain Guests (S-1-5-21-2391453673-3793323726-2791476427-514) -> nobody
Builtin Admins (S-1-5-21-2391453673-3793323726-2791476427-544) -> 510
Builtin Users (S-1-5-21-2391453673-3793323726-2791476427-545) -> 713
```

Объекты LDAP, которые использует idmap (winbind).

```
sambaSidEntry — структурный класс для SID.
sambaIdmapEntry — вспомогательный объект который устанавливает соответствие
между SID и POSIX ID (uid&gid).
```

sambaGroupMapping — вспомогательный класс для мапинга групп.

```
objectclass ( 1.3.6.1.4.1.7165.2.2.4 NAME 'sambaGroupMapping' SUP top AUXILIARY
<------>DESC 'Samba Group Mapping'
<------>MUST ( gidNumber $ sambaSID $ sambaGroupType )
<------>MAY  ( displayName $ description $ sambaSIDList ))
```

```
# BUsers, winsys, Groups, example.com
dn: cn=BUsers,ou=winsys, ou=Groups,dc=example,dc=com
objectClass: top
objectClass: posixGroup
objectClass: sambaGroupMapping
objectClass: groupofuniquenames
cn: Builtin Users
cn: Busers
# 713 — номер posix группы, куда мапится
gidNumber: 713
sambaSID: S-1-5-21-2391453673-3793323726-2791476427-545
sambaSIDList: S-1-5-21-2391453673-3793323726-2791476427-545
sambaGroupType: 4
displayName: Builtin Users
```

```
description: Domain Unix group
uniqueMember: uid=crypt, ou=People, dc=example, dc=com
userPassword:: e2NyeXB0fXg=
```

```
# winsys, Groups, example.com
dn: ou=winsys,ou=Groups, dc=example, dc=com
ou: winsys
description: groups for samba
objectClass: top
objectClass: organizationalunit
objectClass: sambaUnixIdPool
uidNumber: 500
gidNumber: 506

# S-1-5-32-544, winsys, Groups, example.com
dn: sambaSid=S-1-5-32-544, ou=winsys,ou=Groups, dc=example, dc=com
objectClass: sambaSidEntry
objectClass: sambaGroupMapping
objectClass: top
sambaSID: S-1-5-32-544
sambaGroupType: 4
displayName: Administrators
gidNumber: 510
sambaSIDList: S-1-5-21-2391453673-3793323726-2791476427-512

# S-1-5-32-545, winsys, Groups, example.com
dn: sambaSid=S-1-5-32-545, ou=winsys,ou=Groups, dc=example, dc=com
objectClass: sambaSidEntry
objectClass: sambaGroupMapping
objectClass: top
sambaSID: S-1-5-32-545
sambaGroupType: 4
displayName: Users
gidNumber: 99
sambaSIDList: S-1-5-21-2391453673-3793323726-2791476427-513

# S-1-5-21-2391453673-3793323726-2791476427-545, winsys, Groups, example.com
dn: sambasid=S-1-5-21-2391453673-3793323726-2791476427-545, ou=winsys,ou=Group
 s, dc=example, dc=com
gidNumber: 507
objectClass: top
objectClass: sambaidmapentry
objectClass: sambasidentry
sambaSID: S-1-5-21-2391453673-3793323726-2791476427-545
uidNumber: 713

# DAdmins, winsys, Groups, example.com
dn: cn=DAdmins,ou=winsys, ou=Groups,dc=example,dc=com
objectClass: top
objectClass: posixGroup
objectClass: sambaGroupMapping
objectClass: groupofuniquenames
cn: Domain Admins
cn: DAdmins
gidNumber: 510
sambaSID: S-1-5-21-2391453673-3793323726-2791476427-512
sambaSIDList: S-1-5-21-2391453673-3793323726-2791476427-512
sambaGroupType: 4
displayName: Domain Admins
description: Domain Unix group
uniqueMember: uid=crypt, ou=People, dc=example, dc=com
userPassword:: e2NyeXB0fXg=

# DUsers, winsys, Groups, example.com
dn: cn=DUsers,ou=winsys, ou=Groups,dc=example,dc=com
objectClass: top
objectClass: posixGroup
objectClass: sambaGroupMapping
objectClass: groupofuniquenames
cn: Domain Users
cn: DUsers
gidNumber: 510
sambaSID: S-1-5-21-2391453673-3793323726-2791476427-513
sambaSIDList: S-1-5-21-2391453673-3793323726-2791476427-513
sambaGroupType: 4
```

```
displayName: Domain Users
description: Domain Unix group
uniqueMember: uid=crypt, ou=People, dc=example, dc=com
userPassword:: e2NyeXB0fXg=

# DGuests, winsys, Groups, example.com
dn: cn=DGuests,ou=winsys, ou=Groups,dc=example,dc=com
objectClass: top
objectClass: posixGroup
objectClass: sambaGroupMapping
objectClass: groupofuniquenames
cn: Domain Guests
cn: DGuests
gidNumber: 99
sambaSID: S-1-5-21-2391453673-3793323726-2791476427-514
sambaSIDList: S-1-5-21-2391453673-3793323726-2791476427-514
sambaGroupType: 4
displayName: Domain Guests
description: Domain Unix group
uniqueMember: uid=crypt, ou=People, dc=example, dc=com
userPassword:: e2NyeXB0fXg=

# BAdmins, winsys, Groups, example.com
dn: cn=BAdmins,ou=winsys, ou=Groups,dc=example,dc=com
objectClass: top
objectClass: posixGroup
objectClass: sambaGroupMapping
objectClass: groupofuniquenames
cn: Builtin Admins
cn: BAdmins
gidNumber: 510
sambaSID: S-1-5-21-2391453673-3793323726-2791476427-544
sambaSIDList: S-1-5-21-2391453673-3793323726-2791476427-544
sambaGroupType: 4
displayName: Builtin Admins
description: Domain Unix group
uniqueMember: uid=crypt, ou=People, dc=example, dc=com
userPassword:: e2NyeXB0fXg=

# BUsers, winsys, Groups, example.com
dn: cn=BUsers,ou=winsys, ou=Groups,dc=example,dc=com
objectClass: top
objectClass: posixGroup
objectClass: sambaGroupMapping
objectClass: groupofuniquenames
cn: Builtin Users
cn: BUsers
gidNumber: 713
sambaSID: S-1-5-21-2391453673-3793323726-2791476427-545
sambaSIDList: S-1-5-21-2391453673-3793323726-2791476427-545
sambaGroupType: 4
displayName: Builtin Users
description: Domain Unix group
uniqueMember: uid=crypt, ou=People, dc=example, dc=com
userPassword:: e2NyeXB0fXg=
```

Разрешение NetBios имен

Большую сложность представляет разрешение NetBios имен для систем от Microsoft. Если Linux и Mac OS X без проблем работают с Samba сервером с использованием Kerberos аутентификации, то Windows машины, добывая имя машины по Netbios протоколу, отправляют совершенно различные запросы в Kerberos сервер. Так, Windows XP отправлял запросы неизменно делая заглавной первую букву имени хоста. А Windows Vista отправляла запрос со всеми большими. Можно эту проблему решить кардинально — не использовать nmbd, но тогда само собой пропадает Сетевое окружение Windows. В случае с Windows XP можно использовать netbios name, начинающееся с цифры.

Domain Trust Relationship with Windows Domain

Поскольку данный текст стал результатом миграции с домена Windows, то возможность

авторизации пользователей FDS в домене Microsoft и наоборот достаточно важна.

Достигается эта возможность посредством установке доверительных отношений между двумя доменами. Существует различие между Windows Domain и Kerberos Domain (Realm), но в данном случае оно не существенно.

В случае доверительных отношений обмен будет происходить следующим образом:

1. клиент будет посылать стандартный TGT запрос к серверу MIT Kerberos
2. в случае успешного получения, клиент пошлет запрос на получение crossrealm ticket
3. MIT Kerberos пришлет ответ, crossrealm TGT, зашифрованный общим crossrealm ключем
4. клиент использует его для получения доступа в домене MS
5. AD контроллер сопоставит MIT Kerberos account с MS AD account'ом, через account mapping

Как это реализуется

На контроллер домена windows необходимо установить ksetup также, как и на обычную рабочую станцию.

Добавить сервер, отвечающий за MIT kerberos realm

```
C:> ksetup /addkdc EXAMPLE.COM ds.example.com
```

Administrative tools → Active Directory Domains and Trusts

Properties, выбрать **Trust** tab, **Add**

далее пройти конфигуратор использую здравый смысл.

На сервере MIT Kerberos необходимо записи для междоменных отношений

```
addprinc -e des-cbc-crc:normal krbtgt/WINDOM.COM@EXAMPLE.COM
addprinc -e des-cbc-crc:normal krbtgt/EXAMPLE.COM@WINDOM.COM
```

далее необходимо связать учетные записи Windows домена с записями в FDS.

Для этого на AD сервере запускается **Directory Management → Programs → Administrative tools → Active Directory Users and Computers → View → Advanced Features**

Теперь, кликнув на любой аккаунт, можно найти пункт Name Mappings, и далее **Kerberos Names** tab.

Вся эта процедура подробно и с картинками описана на сайте MicroSoft под заголовком Step-by-Step Guide to Kerberos 5 (krb5 1.0) Interoperability.

http://technet.microsoft.com/en-us/library/bb742433.aspx#ECAA

После завершения данной процедуры становится возможным, как авторизироваться на контроллере домена при помощи Kerberos аккаунта, так и использовать AD share folders при помощи smbclient's. То есть smbclient -k -L //share.windom.com должна предоставлять листинг доступных директорий.

Однако с прикладным ПО типа finder на Mac OS X 10.5 возникла проблемы, типа connection failed. Эта ошибка имела место потому, что finder пытался запросить тикет на доступ к несуществующей службе. В то время, как samba правильно запрашивала тикет по hostname$, finder пытался найти cifs/share.example.com.

Squid

В базе Kerberos создается аккаунт и ключ экспортируется на хост web-proxy.

```
addprinc -randkey HTTP/proxy.example.com@EXAMPLE.COM
ktadd -k /root/krb5.keytab HTTP/proxy.example.com@EXAMPLE.COM
scp /root/krb5.keytab proxy.example.com:/etc/squid/HTTP.keytab
```

Аутентификация в squid имеет модульную архитектуру, для поддержки Kerberos используется squid_kerb_auth helper. В наличии этого модуля можно убедиться, если посмотреть /usr/libexec/squid/ или соответстующий путь у вас в системе.

```
rimex ~ # ls /usr/libexec/squid/
cachemgr.cgi     msnt_auth   smb_auth.pl        squid_unix_group
digest_pw_auth   ncsa_auth   smb_auth.sh        unlinkd
diskd            ntlm_auth   squid_kerb_auth    wbinfo_group.pl
fakeauth_auth    pam_auth    squid_ldap_auth
getpwname_auth   sasl_auth   squid_ldap_group
ip_user_check    smb_auth    squid_session
```

У меня с дистрибутивом этого модуля аутентификации не шло, так что потребовалось пересобрать squid с опциями --enable-auth="negotiate" —enable-negotiate-auth-helpers="squid_kerb_auth".

Дополнительные опции в squid.conf

```
auth_param negotiate program /usr/libexec/squid/squid_kerb_auth -d -s
 HTTP/squid_host.example.com@EXAMPLE.COM
# -d - debug
auth_param negotiate children 10
auth_param negotiate keep_alive on
acl authenticated proxy_auth REQUIRED
http_access allow authenticated
```

При старте squid должна экспортироваться

KRB5_KTNAME=/etc/squid/HTTP.keytab

Перечисленых действий достаточно, чтобы все работало.

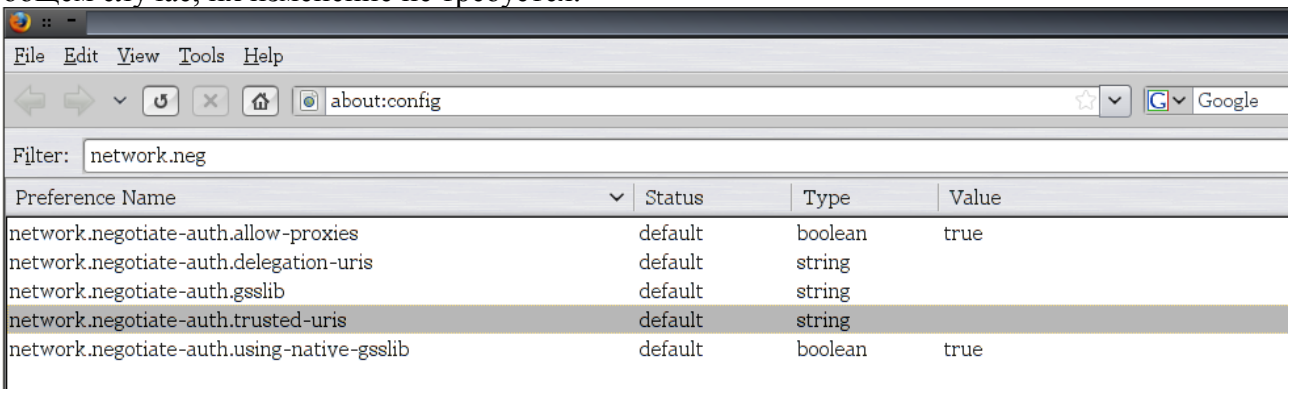
В 'Squid kerberos authentication and ldap authorization in Active Directory' в блоге Klaubert рассказывается про работу с MS Active Directory.

Браузеры

Firefox

Поддержка Kerberos в Firefox осуществляется через GSSAPI в Linux/Unix и SSPI в Windows.

Поддержка в Firefox может быть настроена при помощи следующих опций. Однако в общем случае, их изменение не требуется.



- **network.negotiate-auth.trusted-uris** (default: empty) - For which URLs negotiate authentication should be done
- **network.negotiate-auth.delegation-uris** (default: empty) - For which URLs credential delegation will be allowed - This will in fact give the server side the right to act in your name - WARNING: this requires the client to aquire a forward ticket which is not cached and thus causes a dramatical slow-down of the request.
- **network.negotiate-auth.allow-proxies** (default: true) - Enables proxy authentication using the negotiate method / The Squid guys want to support it in version 3.
- **network.negotiate-auth.using-native-gsslib** - Use the default GSSAPI library (does not mean SSPI on windows)
- **network.negotiate-auth.gsslib** (default:empty) - Specifies a alternate GSSAPI shared library

- **network.auth.use-sspi** (only on Windows, default: true) - Whether to use Microsoft's SSPI library, if disabled use GSSAPI

Взято из Doing GSS/Negotiate SSO using Mozilla Firefox, MIT Kerberos and PHP.

MS Internet Explorer

Опция 'Integrated Windows Authentication'

Safari

Поддерживает без каких либо действий. (утверждение нуждается в тестировании)

Opera

Текущая версия 9.5 не поддерживает Kerberos

Замечание об адресной книге

Известно, что адресную книгу из LDAP могу брать практически все почтовые клиенты. При этом есть набор базовых объектов, описанных в RFC. И есть недокументированные расширения от Microsoft, Mozilla и т.д. В Приложении есть схемы, которые я использовал. В случае большого разнообразия почтовых клиентов организация общей адресной книги может быть оказаться не такой уж хорошей идеей.

Вместо заключения

Таким образом, при должном упорстве и навыках можно создать структуру на основе Open Source для централизованного управления пользователями и ресурсами. Очевидно, что у неё есть сильные стороны, готовые ко внедрению. Однако целостность интеграции полностью ложиться на плечи администратора сети. И я надеюсь, этот документ поможет при планировании подобного рода систем.

Приложение

ol-schema-migrate.pl - скрипт для конвертации openldap схем для использования в FDS

```
#!/usr/bin/perl -w
#
# Convert OpenLDAP schema files into Fedora DS format with RFC2252 compliant printing
#
# First Release : Mike Jackson <mj@sci.fi> 14 June 2005
#    http://www.netauth.com/~jacksonm/ldap/ol-schema-migrate.pl
#    Professional LDAP consulting for large and small projects
#
# - 6 Dec 2005
# - objectclass element ordering
#
# Second Release : Alyseo <info@alyseo.com> 05 Februrary 2006
#    Francois Billard <francois@alyseo.com>
#    Yacine Kheddache <yacine@alyseo.com>
#    http://www.alyseo.com
#
# - 05 Februrary 2006
#  - parsing improvements to accept schema not RFC compliant like ISPMAN
#  - adding RFC element : Usage, No-user-modification, collective keywords
# - 08 Februrary 2006
#  - adding help & usage
#  - now it can beautify your schemas: "-b"
#  - count attributes and objects class: "-c"
#  - display items that can not be converted (empty OID...): "-d"
# - 15 February 2006
#  - adding workaround for Fedora DS bug 181465:
#    https://bugzilla.redhat.com/bugzilla/show_bug.cgi?id=181465
#  - adding duplicated OID check: "-d"
#    Useful to manually correct nasty schemas like:
#    https://sourceforge.net/tracker/?func=detail&atid=108390&aid=1429276&group_id=8390
#
# - Fedora DS bug you need to correct by hang (this script is not taking it into account):
```

```
#    https://bugzilla.redhat.com/bugzilla/show_bug.cgi?id=179956
#
# GPL license
#

my $optionCount = 0;
my $optionPrint = 0;
my $optionBadEntries = 0;
my $optionHelp = 0;
my $filename = "" ;

foreach (@ARGV) {
  $optionHelp = 1 if ( /^-h$/);
  $optionCount = 1 if ( /^-c$/);
  $optionPrint = 1 if ( /^-b$/);
  $optionBadEntries = 1 if ( /^-d$/);
  $filename = $_ if ( ! /^-b$/ && ! /^-c$/ && ! /^-d$/);
}

die "Usage : ol-schema-migrate-v2.pl [ -c ] [ -b ] [ -d ] schema\n" .
    "  -c\tcount attribute and object class\n" .
    "  -b\tconvert and beautify your schema\n" .
    "  -d\tdisplay unrecognized elements, find empty and duplicated OID\n" .
    "  -h\tthis help\n" if ($filename eq "" || ($optionHelp || (!$optionCount && !$optionPrint && !
$optionBadEntries)));

if($optionCount) {
  print "Schema verification counters:\n";
  my $ldapdata = &getSourceFile($filename);
  print "".(defined($ldapdata->{attributes}) ? @{$ldapdata->{attributes}} : 0) . " attributes\n";
  print "".(defined($ldapdata->{objectclass}) ?  @{$ldapdata->{objectclass}} : 0) . " object
classes\n\n"
}

if($optionPrint) {
  my $ldapdata = &getSourceFile($filename);
  &printit($ldapdata);
}

if($optionBadEntries) {
  print "Display unrecognized entries:\n";
  my $ldapdata = &getSourceFile($filename);
  my $errorsAttr = 0;
  my $errorsObjc = 0;
  my $errorsDup  = 0;
  my $emptyOid = 0;
  my %dup;

  foreach (@{$ldapdata->{attributes}}) {
    my $attr = $_;

    push @{$dup{$attr->{OID}}{attr}}, {NAME => $attr->{NAME}, LINENUMBER => $attr->{LINENUMBER}};

    $attr->{DATA} =~ s/\n/ /g;
    $attr->{DATA} =~ s/\r//g;
    $attr->{DATA} =~ s/attribute[t|T]ypes?:?\s*\(//;
    $attr->{DATA} =~ s/\Q$attr->{OID}//                  if(defined $attr->{OID});
    $attr->{DATA} =~ s/NAME\s*\Q$attr->{NAME}//          if(defined $attr->{NAME});
    $attr->{DATA} =~ s/DESC\s*'\Q$attr->{DESC}'//        if(defined $attr->{DESC});
    $attr->{DATA} =~ s/$attr->{OBSOLETE}//               if(defined $attr->{OBSOLETE});
    $attr->{DATA} =~ s/SUP\s*\Q$attr->{SUP}//            if(defined $attr->{SUP});
    $attr->{DATA} =~ s/EQUALITY\s*\Q$attr->{EQUALITY}// if(defined $attr->{EQUALITY});
    $attr->{DATA} =~ s/ORDERING\s*\Q$attr->{ORDERING}// if(defined $attr->{ORDERING});
    $attr->{DATA} =~ s/SUBSTR\s*\Q$attr->{SUBSTR}//      if(defined $attr->{SUBSTR});
    $attr->{DATA} =~ s/SYNTAX\s*\Q$attr->{SYNTAX}//      if(defined $attr->{SYNTAX});
    $attr->{DATA} =~ s/SINGLE-VALUE//                    if(defined $attr->{SINGLEVALUE});
    $attr->{DATA} =~ s/NO-USER-MODIFICATION//            if(defined $attr->{NOUSERMOD});
    $attr->{DATA} =~ s/COLLECTIVE//                      if(defined $attr->{COLLECTIVE});
    $attr->{DATA} =~ s/USAGE\s*\Q$attr->{USAGE}//        if(defined $attr->{USAGE});
    $attr->{DATA} =~ s/\)\s$//;
    $attr->{DATA} =~ s/^\s+(\S)/\n$1/  ;
    $attr->{DATA} =~ s/(\S)\s+$/$1\n/;
    do {
      $errorsAttr ++;
      do {  $emptyOid ++;
            print "Warning : no OID for attributes element at line $attr->{LINENUMBER} \n";
```

```
      } if( !defined($attr->{OID}));
      print "### Unknow element embedded in ATTRIBUTE at line $attr->{LINENUMBER} :\n$attr->{DATA}\
n"
    } if($attr->{DATA} =~ /\w/);
  }

  foreach (@{$ldapdata->{objectclass}}) {
    my $objc = $_;
    push @{$dup{$objc->{OID}}{objc}} , {NAME => $objc->{NAME}, LINENUMBER => $objc->{LINENUMBER}};
    $objc->{DATA} =~ s/\n/ /g;
    $objc->{DATA} =~ s/\r//g;
    $objc->{DATA} =~ s/^object[c|C]lasse?s?:?\s*\(?//;
    $objc->{DATA} =~ s/\Q$objc->{OID}//                       if(defined $objc->{OID});
    $objc->{DATA} =~ s/NAME\s*\Q$objc->{NAME}\E//             if(defined $objc->{NAME});
    $objc->{DATA} =~ s/DESC\s*'\Q$objc->{DESC}\E'//           if(defined $objc->{DESC});
    $objc->{DATA} =~ s/OBSOLETE//                             if(defined $objc->{OBSOLETE});
    $objc->{DATA} =~ s/SUP\s*\Q$objc->{SUP}//                 if(defined $objc->{SUP});
    $objc->{DATA} =~ s/\Q$objc->{TYPE}//                      if(defined $objc->{TYPE});
    $objc->{DATA} =~ s/MUST\s*\Q$objc->{MUST}\E\s*//          if(defined $objc->{MUST});
    $objc->{DATA} =~ s/MUST\s*\(?\s*\Q$objc->{MUST}\E\s*\)?// if(defined $objc->{MUST});
    $objc->{DATA} =~ s/MAY\s*\Q$objc->{MAY}\E//               if(defined $objc->{MAY});
    $objc->{DATA} =~ s/\)\s$//;
    $objc->{DATA} =~ s/^\s+(\S)/\n$1/  ;
    $objc->{DATA} =~ s/(\S)\s+$/$1\n/;

    do {
      print "#" x 80 ."\n";
      $errorsObjc ++;
      do { $emptyOid++ ;
           print "Warning : no OID for object class element at line $objc->{LINENUMBER} \n";
      } if( $objc->{OID} eq "");
      print "### Unknow element embedded in OBJECT CLASS at line $objc->{LINENUMBER} :\n$objc-
>{DATA}\n"
    } if($objc->{DATA} =~ /\w/);
  }

  my $nbDup = 0;
  foreach (keys %dup) {
    my $sumOid = 0;
    $sumOid += @{$dup{$_}{attr}} if(defined (@{$dup{$_}{attr}}));
    $sumOid += @{$dup{$_}{objc}} if(defined (@{$dup{$_}{objc}}));
    if( $sumOid > 1 && $_ ne "") {
      $nbDup ++;
      print "#" x 80 ."\n";
      print "Duplicate OID founds : $_\n";
      foreach (@{$dup{$_}{attr}}) {

        print "Attribute : $_->{NAME} (line : $_->{LINENUMBER})\n";
      }
      foreach (@{$dup{$_}{objc}}) {
        print "Object class : $_->{NAME} (line : $_->{LINENUMBER})\n";
      }

    }
  }

  print "\n$errorsAttr errors detected in ATTRIBUTES list\n";
  print "$errorsObjc errors detected in OBJECT CLASS list\n";
  print "$nbDup duplicate OID founds\n";
  print "$emptyOid empty OID fields founds\n\n";

}


sub printit {
  my $ldapdata = shift;
  &printSeparator;
  print "dn: cn=schema\n";
  &printSeparator;

  # print elements in RFC2252 order

  foreach (@{$ldapdata->{attributes}}) {
    my $attr = $_;
    print "attributeTypes: (\n";
    print "  $attr->{OID}\n";
```

```
    print "  NAME $attr->{NAME}\n";
    print "  DESC '$attr->{DESC}'\n"         if(defined $attr->{DESC});
    print "  OBSOLETE\n"                     if(defined $attr->{OBSOLETE});
    print "  SUP $attr->{SUP}\n"             if(defined $attr->{SUP});
    print "  EQUALITY $attr->{EQUALITY}\n"   if(defined $attr->{EQUALITY});
    print "  ORDERING $attr->{ORDERING}\n"   if(defined $attr->{ORDERING});
    print "  SUBSTR $attr->{SUBSTR}\n"       if(defined $attr->{SUBSTR});
    print "  SYNTAX $attr->{SYNTAX}\n"       if(defined $attr->{SYNTAX});
    print "  SINGLE-VALUE\n"                 if(defined $attr->{SINGLEVALUE});
    print "  NO-USER-MODIFICATION\n"         if(defined $attr->{NOUSERMOD});
    print "  COLLECTIVE\n"                   if(defined $attr->{COLLECTIVE});
    print "  USAGE $attr->{USAGE}\n"         if(defined $attr->{USAGE});
    print "  )\n";
    &printSeparator;
  }

  foreach (@{$ldapdata->{objectclass}}) {
    my $objc = $_;
    # next 3 lines : Fedora DS space sensitive bug workaround
    $objc->{SUP}         =~ s/^\(\s*(.*?)\s*\)$/\( $1 \)/  if (defined $objc->{SUP});
    $objc->{MUST}        =~ s/^\(\s*(.*?)\s*\)$/\( $1 \)/  if (defined $objc->{MUST});
    $objc->{MAY}         =~ s/^\(\s*(.*?)\s*\)$/\( $1 \)/  if (defined $objc->{MAY});

    print "objectClasses: (\n";
    print "  $objc->{OID}\n";
    print "  NAME $objc->{NAME}\n";
    print "  DESC '$objc->{DESC}'\n"  if(defined $objc->{DESC});
    print "  OBSOLETE\n"              if(defined $objc->{OBSOLETE});
    print "  SUP $objc->{SUP}\n"      if(defined $objc->{SUP});
    print "  $objc->{TYPE}\n"         if(defined $objc->{TYPE});
    print "  MUST $objc->{MUST}\n"    if(defined $objc->{MUST});
    print "  MAY $objc->{MAY}\n"      if(defined $objc->{MAY});
    print "  )\n";
    &printSeparator;
  }
}

sub printSeparator {
  print "#\n";
  print "#" x 80 . "\n";
  print "#\n";
}

sub getSourceFile {
  my @data = &getFile(shift);
  my %result;
  my $result = \%result;
  my @allattrs;
  my @allattrsLineNumber;
  my @allobjc;
  my @allobjcLineNumber;
  my $at = 0;
  my $oc = 0;
  my $at_string;
  my $oc_string;
  my $idx = 0;
  my $beginParenthesis = 0;
  my $endParenthesis = 0;
  my $lineNumber = 0;
  for(@data) {
    $lineNumber++;
    next if (/^\s*\#/); # skip comments

    if($at) {
      s/ +/ /;                    # remove embedded tabs
      s/\t/ /;                    # remove multiple spaces after the $ sign

       $at_string .= $_;
      $beginParenthesis = 0;      # Use best matching elements
      $endParenthesis = 0;
      for(my $i=0;$ i < length($at_string); $i++) {
        $beginParenthesis++ if(substr ($at_string,$i,1) eq "(");
        $endParenthesis++ if(substr ($at_string,$i,1) eq ")");
      }
      if($beginParenthesis == $endParenthesis) {
        push @allattrs, $at_string;
```

```
        $at = 0;
        $at_string = "";
        $endParenthesis = 0;
        $beginParenthesis = 0;
      }
    }

    if (/^attribute[t|T]ype/) {
      my $line = $_;
       push @allattrsLineNumber, $lineNumber;      # keep starting line number
      for(my $i=0;$ i < length($line); $i++) {
        $beginParenthesis++ if(substr ($line, $i, 1) eq "(");
        $endParenthesis++ if(substr ($line, $i, 1) eq ")");
      }
      if($beginParenthesis == $endParenthesis && $beginParenthesis != 0) {
        push @allattrs, $line;
        $endParenthesis = 0;
        $beginParenthesis = 0;
      } else {
        $at_string = $line;
        $at = 1;
      }
    }

    ####################################

    if($oc) {
      s/ +/ /;
      s/\t/ /;

      $oc_string .= $_;
      $endParenthesis = 0;          # best methode to accept an elements :
      $beginParenthesis = 0;        # left parenthesis sum == right parenthesis sum, so we are sure
to
      for(my $i=0;$ i < length($oc_string); $i++) {      # have an element.
        $beginParenthesis++ if(substr ($oc_string, $i, 1) eq "(");
        $endParenthesis++ if(substr ($oc_string, $i, 1) eq ")");
      }
      if($beginParenthesis == $endParenthesis) {
        push @allobjc, $oc_string;
        $oc = 0;
        $oc_string = "";
        $endParenthesis = 0;
        $beginParenthesis = 0;
      }
    }

    if (/^object[c|C]lass/) {
      my $line = $_;
       push @allobjcLineNumber, $lineNumber;    # keep starting line number
      for(my $i=0;$ i < length($line); $i++) {
        $beginParenthesis++ if(substr ($line, $i, 1) eq "(");
        $endParenthesis++ if(substr ($line, $i, 1) eq ")");
      }
      if($beginParenthesis == $endParenthesis && $beginParenthesis != 0) {
        push @allobjc, $line;
        $endParenthesis = 0;
        $beginParenthesis = 0;
      } else {
        $oc_string = $line;
        $oc = 1;
      }
    }
  }

  # Parsing attribute elements

  for(@allattrs) {
    s/\n/ /g;
    s/\r//g;
    s/ +/ /g;
    s/\t/ /g;
    $result->{attributes}->[$idx]->{DATA}        = $_              if($optionBadEntries);     #
keep original data
    $result->{attributes}->[$idx]->{LINENUMBER}  = $allattrsLineNumber[$idx];
    $result->{attributes}->[$idx]->{OID}         = $1              if (m/^attribute[t|T]ypes?:?\s*\
```

```
(?\s*([\.\d]*?)\s+/);
    $result->{attributes}->[$idx]->{NAME}        = $1              if (m/NAME\s+('.*?')\s*/ ||
m/NAME\s+(\(.*?\))/);
    $result->{attributes}->[$idx]->{DESC}        = $1              if (m/DESC\s+'(.*?)'\s*/);
    $result->{attributes}->[$idx]->{OBSOLETE}    = "OBSOLETE"      if (m/OBSOLETE/);
    $result->{attributes}->[$idx]->{SUP}         = $1              if (m/SUP\s+(.*?)\s/);
    $result->{attributes}->[$idx]->{EQUALITY}    = $1              if (m/EQUALITY\s+(.*?)\s/);
    $result->{attributes}->[$idx]->{ORDERING}    = $1              if (m/ORDERING\s+(.*?)\s/);
    $result->{attributes}->[$idx]->{SUBSTR}      = $1              if (m/SUBSTR\s+(.*?)\s/);
    $result->{attributes}->[$idx]->{SYNTAX}      = $1              if (m/SYNTAX\s+(.*?)(\s|\))/);
    $result->{attributes}->[$idx]->{SINGLEVALUE} = "SINGLE-VALUE"  if (m/SINGLE-VALUE/);
    $result->{attributes}->[$idx]->{COLLECTIVE}  = "COLLECTIVE"    if (m/COLLECTIVE/);
    $result->{attributes}->[$idx]->{USAGE}       = $1              if (m/USAGE\s+(.*?)\s/);
    $result->{attributes}->[$idx]->{NOUSERMOD}   = "NO-USER-MODIFICATION"    if (m/NO-USER-
MODIFICATION/);
    $idx ++;
  }

  $idx = 0;

  # Parsing object class elements

  for(@allobjc) {
    s/\n/ /g;
    s/\r//g;
    s/ +/ /g;
    s/\t/ /g;
    $result->{objectclass}->[$idx]->{DATA}        = $_          if($optionBadEntries);     # keep
original data
    $result->{objectclass}->[$idx]->{LINENUMBER}  = $allobjcLineNumber[$idx];
    $result->{objectclass}->[$idx]->{OID}         = $1          if (m/^object[c|C]lasse?s?:?\s*\(?\
s*([\.\d]*?)\s+/);
    $result->{objectclass}->[$idx]->{NAME}        = $1          if (m/NAME\s+('.*?')\s*/ || m/NAME\
s+(\(.*?\))/);
    $result->{objectclass}->[$idx]->{DESC}        =  $1         if (m/DESC\s+'(.*?)'\s*/);
    $result->{objectclass}->[$idx]->{OBSOLETE}    = "OBSOLETE"  if (m/OBSOLETE/);
    $result->{objectclass}->[$idx]->{SUP}         = $1          if (m/SUP\s+([^()]+?)\s/ ||
m/SUP\s+(\(.+?\))\s/);
    $result->{objectclass}->[$idx]->{TYPE}        = $1          if (m/((?:STRUCTURAL)|
(?:AUXILIARY)|(?:ABSTRACT))/);
    $result->{objectclass}->[$idx]->{MUST}        = $1          if (m/MUST\s+(\w+)\)?/ || m/MUST\s+
(\(.*?\))(\s|\))/s);
    $result->{objectclass}->[$idx]->{MAY}         = $1          if (m/MAY\s+(\w+)\)?/ || m/MAY\s+(\
(.*?\))(\s|\))/s);

    $idx++;
  }

  return $result;
}

sub getFile {
  my @data;
  my $file = shift;
  die "File not found : $file\n" if(! -e $file);
  open FH, $file;
  @data = <FH>;
  close FH;
  @data;
}
```

86sudo.ldif

```
dn: cn=schema
attributeTypes: ( 2.16.840.1.113719.1.301.4.1.1
                NAME 'krbPrincipalName'
                EQUALITY caseExactIA5Match
        SUBSTR caseExactSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
attributeTypes: ( 2.16.840.1.113719.1.301.4.3.1
                NAME 'krbPrincipalType'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.5.1
                NAME 'krbUPEnabled'
```

```
                DESC 'Boolean'
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.6.1
                NAME 'krbPrincipalExpiration'
                EQUALITY generalizedTimeMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.8.1
                NAME 'krbTicketFlags'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.9.1
                NAME 'krbMaxTicketLife'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.10.1
                NAME 'krbMaxRenewableAge'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.14.1
                NAME 'krbRealmReferences'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.15.1
                NAME 'krbLdapServers'
                EQUALITY caseIgnoreMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)
attributeTypes: ( 2.16.840.1.113719.1.301.4.17.1
                NAME 'krbKdcServers'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.18.1
                NAME 'krbPwdServers'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.24.1
                NAME 'krbHostServer'
                EQUALITY caseExactIA5Match
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
attributeTypes: ( 2.16.840.1.113719.1.301.4.25.1
                NAME 'krbSearchScope'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.26.1
                NAME 'krbPrincipalReferences'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.28.1
                NAME 'krbPrincNamingAttr'
                EQUALITY caseIgnoreMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.29.1
                NAME 'krbAdmServers'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.30.1
                NAME 'krbMaxPwdLife'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.31.1
                NAME 'krbMinPwdLife'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.32.1
                NAME 'krbPwdMinDiffChars'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
```

```
attributeTypes: ( 2.16.840.1.113719.1.301.4.33.1
                NAME 'krbPwdMinLength'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.34.1
                NAME 'krbPwdHistoryLength'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.36.1
                NAME 'krbPwdPolicyReference'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.37.1
                NAME 'krbPasswordExpiration'
                EQUALITY generalizedTimeMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.39.1
                NAME 'krbPrincipalKey'
                EQUALITY octetStringMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.40)
attributeTypes: ( 2.16.840.1.113719.1.301.4.40.1
                NAME 'krbTicketPolicyReference'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.41.1
                NAME 'krbSubTrees'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.42.1
                NAME 'krbDefaultEncSaltTypes'
                EQUALITY caseIgnoreMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)
attributeTypes: ( 2.16.840.1.113719.1.301.4.43.1
                NAME 'krbSupportedEncSaltTypes'
                EQUALITY caseIgnoreMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)
attributeTypes: ( 2.16.840.1.113719.1.301.4.44.1
                NAME 'krbPwdHistory'
                EQUALITY octetStringMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.40)
attributeTypes: ( 2.16.840.1.113719.1.301.4.45.1
                NAME 'krbLastPwdChange'
                EQUALITY generalizedTimeMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.46.1
                NAME 'krbMKey'
                EQUALITY octetStringMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.40)
attributeTypes: ( 2.16.840.1.113719.1.301.4.47.1
                NAME 'krbPrincipalAliases'
                EQUALITY caseExactIA5Match
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
attributeTypes: ( 2.16.840.1.113719.1.301.4.48.1
                NAME 'krbLastSuccessfulAuth'
                EQUALITY generalizedTimeMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.49.1
                NAME 'krbLastFailedAuth'
                EQUALITY generalizedTimeMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.50.1
                NAME 'krbLoginFailedCount'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.51.1
                NAME 'krbExtraData'
                EQUALITY octetStringMatch
```

```
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.40)
attributeTypes: ( 2.16.840.1.113719.1.301.4.52.1
                NAME 'krbObjectReferences'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.53.1
                NAME 'krbPrincContainerRef'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
objectclasses: ( 2.16.840.1.113719.1.301.6.1.1 NAME 'krbContainer' SUP top STRUCTURAL MUST ( cn ) )
objectclasses: ( 2.16.840.1.113719.1.301.6.2.1 NAME 'krbRealmContainer' SUP top STRUCTURAL MUST
( cn )
                MAY ( krbMKey $ krbUPEnabled $ krbSubTrees $ krbSearchScope $ krbLdapServers $
krbSupportedEncSaltTypes $ krbDefaultEncSaltTypes $ krbTicketPolicyReference $ krbKdcServers $
krbPwdServers $ krbAdmServers $ krbPrincNamingAttr $ krbPwdPolicyReference $ krbPrincContainerRef )
)
objectclasses: ( 2.16.840.1.113719.1.301.6.3.1
                NAME 'krbService'
                SUP top
                ABSTRACT
                MUST ( cn )
                MAY ( krbHostServer $ krbRealmReferences ) )
objectclasses: ( 2.16.840.1.113719.1.301.6.4.1 NAME 'krbKdcService' SUP krbService STRUCTURAL )
objectclasses: ( 2.16.840.1.113719.1.301.6.5.1
                NAME 'krbPwdService' SUP krbService STRUCTURAL )
objectclasses: ( 2.16.840.1.113719.1.301.6.8.1
                NAME 'krbPrincipalAux'
        SUP top
                AUXILIARY
                MAY ( krbPrincipalName $ krbUPEnabled $ krbPrincipalKey $ krbTicketPolicyReference
$ krbPrincipalExpiration $ krbPasswordExpiration $ krbPwdPolicyReference $ krbPrincipalType $
krbPwdHistory $ krbLastPwdChange $ krbPrincipalAliases $ krbLastSuccessfulAuth $ krbLastFailedAuth
$ krbLoginFailedCount $ krbExtraData ) )
objectclasses: ( 2.16.840.1.113719.1.301.6.9.1
                NAME 'krbPrincipal'
                SUP top
                MUST ( krbPrincipalName )
        MAY ( krbObjectReferences ) )
objectclasses: ( 2.16.840.1.113719.1.301.6.11.1
                NAME 'krbPrincRefAux'
                SUP top
                AUXILIARY
                MAY krbPrincipalReferences )
objectclasses: ( 2.16.840.1.113719.1.301.6.13.1
                NAME 'krbAdmService'
                SUP krbService
        STRUCTURAL )
objectclasses: ( 2.16.840.1.113719.1.301.6.14.1
                NAME 'krbPwdPolicy'
                SUP top
                MUST ( cn )
                MAY ( krbMaxPwdLife $ krbMinPwdLife $ krbPwdMinDiffChars $ krbPwdMinLength $
krbPwdHistoryLength ) )
objectclasses: ( 2.16.840.1.113719.1.301.6.16.1
                NAME 'krbTicketPolicyAux'
                SUP top
                AUXILIARY
                MAY ( krbTicketFlags $ krbMaxTicketLife $ krbMaxRenewableAge ) )
objectclasses: ( 2.16.840.1.113719.1.301.6.17.1
                NAME 'krbTicketPolicy'
                SUP top
                MUST ( cn ) )
```

87mozilla.ldif

```
#
#############################################################################
#
dn: cn=schema
#
#############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.4.1
  NAME 'mozillaCustom1'
  EQUALITY caseIgnoreMatch
```

```
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128}
  SINGLE-VALUE
  )
#
###############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.4.2
  NAME 'mozillaCustom2'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128}
  SINGLE-VALUE
  )
#
###############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.4.3
  NAME 'mozillaCustom3'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128}
  SINGLE-VALUE
  )
#
###############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.4.4
  NAME 'mozillaCustom4'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128}
  SINGLE-VALUE
  )
#
###############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.3.1
  NAME 'mozillaHomeStreet'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128}
  SINGLE-VALUE
  )
#
###############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.3.2
  NAME 'mozillaHomeStreet2'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128}
  SINGLE-VALUE
  )
#
###############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.3.3
  NAME 'mozillaHomeLocalityName'
  SUP name
  SINGLE-VALUE
  )
#
###############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.3.4
  NAME 'mozillaHomeState'
  SUP name
  SINGLE-VALUE
```

```
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.3.5
  NAME 'mozillaHomePostalCode'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{40}
  SINGLE-VALUE
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.3.6
  NAME 'mozillaHomeCountryName'
  SUP name
  SINGLE-VALUE
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.3.7
  NAME 'mozillaHomeUrl'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128}
  SINGLE-VALUE
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.3.8
  NAME 'mozillaWorkStreet2'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128}
  SINGLE-VALUE
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.3.9
  NAME 'mozillaWorkUrl'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128}
  SINGLE-VALUE
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.2.1
  NAME ( 'mozillaNickname' 'xmozillanickname' )
  SUP name
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.2.2
  NAME ( 'mozillaSecondEmail' 'xmozillasecondemail' )
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256}
  SINGLE-VALUE
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.13769.2.3
```

```
  NAME ( 'mozillaUseHtmlMail' 'xmozillausehtmlmail' )
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
  SINGLE-VALUE
  )
#
###############################################################################
#
objectClasses: (
  1.3.6.1.4.1.13769.9.1
  NAME 'mozillaAbPersonAlpha'
  SUP top
  AUXILIARY
  MUST ( cn )
  MAY ( c $  description $ displayName $ fax $ givenName $ homePhone $ l $ mail $ mobile $
mozillaCustom1 $ mozillaCustom2 $ mozillaCustom3 $ mozillaCustom4 $ mozillaHomeCountryName $
mozillaHomeLocalityName $ mozillaHomePostalCode $ mozillaHomeState $ mozillaHomeStreet $
mozillaHomeStreet2 $ mozillaHomeUrl $ mozillaNickname $ mozillaSecondEmail $ mozillaUseHtmlMail $
mozillaWorkStreet2 $ mozillaWorkUrl $ o $ ou $ pager $ postalCode $ postOfficeBox $ sn $ st $
street $ telephoneNumber $  title )
  )
#
###############################################################################
#
```

88evolutionperson.ldif

```
dn: cn=schema
attributeTypes: ( 1.3.6.1.4.1.8506.1.2.1
  NAME 'primaryPhone'
  DESC 'preferred phone number used to contact a person'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
  )
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.2
  NAME 'carPhone'
  DESC 'car phone telephone number of the person'
  EQUALITY telephoneNumberMatch
  SUBSTR telephoneNumberSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50
  SINGLE-VALUE
  )
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.4
  NAME 'otherPhone'
  EQUALITY telephoneNumberMatch
  SUBSTR telephoneNumberSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50
  )
#
###############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.5
  NAME 'businessRole'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  )
#
###############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.6
  NAME 'managerName'
  SUP name
  )
#
###############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.7
  NAME 'assistantName'
  SUP name
```

```
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.8
  NAME 'spouseName'
  SUP name
  SINGLE-VALUE
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.9
  NAME 'otherPostalAddress'
  EQUALITY caseIgnoreListMatch
  SUBSTR caseIgnoreListSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.41
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.10
  NAME ( 'mailer' 'mua' )
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32}
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.11
  NAME ( 'birthDate' 'dob' )
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128}
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.12
  NAME 'anniversary'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{128}
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.13
  NAME 'note'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{1024}
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.14
  NAME 'evolutionArbitrary'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{4096}
  )
#
#########################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.15
  NAME 'fileAs'
```

```
  SUP name
  )
#
##############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.16
  NAME 'assistantPhone'
  EQUALITY telephoneNumberMatch
  SUBSTR telephoneNumberSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50
  )
#
##############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.17
  NAME 'companyPhone'
  EQUALITY telephoneNumberMatch
  SUBSTR telephoneNumberSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50
  )
#
##############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.18
  NAME 'callbackPhone'
  EQUALITY telephoneNumberMatch
  SUBSTR telephoneNumberSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50
  )
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.20
  NAME 'radio'
  EQUALITY telephoneNumberMatch
  SUBSTR telephoneNumberSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50
  )
#
##############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.21
  NAME 'telex'
  EQUALITY telephoneNumberMatch
  SUBSTR telephoneNumberSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50
  )
#
##############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.22
  NAME 'tty'
  EQUALITY telephoneNumberMatch
  SUBSTR telephoneNumberSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50
  )
#
##############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.23
  NAME 'categories'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{4096}
  )
#
##############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.24
  NAME 'contact'
  EQUALITY distinguishedNameMatch
```

```
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
  )
#
#############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.25
  NAME 'listName'
  SUP name
  SINGLE-VALUE
  )
#
#############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.26
  NAME 'calendarURI'
  EQUALITY caseExactIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
  )
#
#############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.27
  NAME 'freeBusyURI'
  EQUALITY caseExactIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
  )
#
#############################################################################
#
attributeTypes: (
  1.3.6.1.4.1.8506.1.2.28
  NAME 'category'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{4096}
  )
#
#############################################################################
#
objectClasses: (
  1.3.6.1.4.1.8506.1.3.1
  NAME 'evolutionPerson'
  DESC 'Objectclass geared to Evolution Usage'
  SUP inetOrgPerson
  STRUCTURAL
  MAY ( fileAs $ primaryPhone $ carPhone $ otherPhone $ businessRole $ managerName $ assistantName
$ assistantPhone $  otherPostalAddress $ mailer $ birthDate $ anniversary $ spouseName $  note $
companyPhone $ callbackPhone $ radio $ telex $ tty $ categories $ category $ calendarURI $
freeBusyURI )
  )
#
#############################################################################
#
objectClasses: (
  1.3.6.1.4.1.8506.1.3.2
  NAME 'evolutionPersonList'
  DESC 'Objectclass geared to Evolution Contact Lists'
  SUP top
  STRUCTURAL
  MUST ( listName )
  MAY ( mail $ contact )
  )
#
#############################################################################
#
```

89kerberos.ldif

```
dn: cn=schema
attributeTypes: ( 2.16.840.1.113719.1.301.4.1.1
                NAME 'krbPrincipalName'
```

```
                EQUALITY caseExactIA5Match
        SUBSTR caseExactSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
attributeTypes: ( 2.16.840.1.113719.1.301.4.3.1
                NAME 'krbPrincipalType'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.5.1
                NAME 'krbUPEnabled'
                DESC 'Boolean'
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.6.1
                NAME 'krbPrincipalExpiration'
                EQUALITY generalizedTimeMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.8.1
                NAME 'krbTicketFlags'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.9.1
                NAME 'krbMaxTicketLife'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.10.1
                NAME 'krbMaxRenewableAge'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.14.1
                NAME 'krbRealmReferences'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.15.1
                NAME 'krbLdapServers'
                EQUALITY caseIgnoreMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)
attributeTypes: ( 2.16.840.1.113719.1.301.4.17.1
                NAME 'krbKdcServers'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.18.1
                NAME 'krbPwdServers'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.24.1
                NAME 'krbHostServer'
                EQUALITY caseExactIA5Match
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
attributeTypes: ( 2.16.840.1.113719.1.301.4.25.1
                NAME 'krbSearchScope'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.26.1
                NAME 'krbPrincipalReferences'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.28.1
                NAME 'krbPrincNamingAttr'
                EQUALITY caseIgnoreMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.29.1
                NAME 'krbAdmServers'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.30.1
                NAME 'krbMaxPwdLife'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
```

```
attributeTypes: ( 2.16.840.1.113719.1.301.4.31.1
                NAME 'krbMinPwdLife'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.32.1
                NAME 'krbPwdMinDiffChars'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.33.1
                NAME 'krbPwdMinLength'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.34.1
                NAME 'krbPwdHistoryLength'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.36.1
                NAME 'krbPwdPolicyReference'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.37.1
                NAME 'krbPasswordExpiration'
                EQUALITY generalizedTimeMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.39.1
                NAME 'krbPrincipalKey'
                EQUALITY octetStringMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.40)
attributeTypes: ( 2.16.840.1.113719.1.301.4.40.1
                NAME 'krbTicketPolicyReference'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.41.1
                NAME 'krbSubTrees'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.42.1
                NAME 'krbDefaultEncSaltTypes'
                EQUALITY caseIgnoreMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)
attributeTypes: ( 2.16.840.1.113719.1.301.4.43.1
                NAME 'krbSupportedEncSaltTypes'
                EQUALITY caseIgnoreMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)
attributeTypes: ( 2.16.840.1.113719.1.301.4.44.1
                NAME 'krbPwdHistory'
                EQUALITY octetStringMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.40)
attributeTypes: ( 2.16.840.1.113719.1.301.4.45.1
                NAME 'krbLastPwdChange'
                EQUALITY generalizedTimeMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.46.1
                NAME 'krbMKey'
                EQUALITY octetStringMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.40)
attributeTypes: ( 2.16.840.1.113719.1.301.4.47.1
                NAME 'krbPrincipalAliases'
                EQUALITY caseExactIA5Match
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
attributeTypes: ( 2.16.840.1.113719.1.301.4.48.1
                NAME 'krbLastSuccessfulAuth'
                EQUALITY generalizedTimeMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.49.1
                NAME 'krbLastFailedAuth'
                EQUALITY generalizedTimeMatch
```

```
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.50.1
                NAME 'krbLoginFailedCount'
                EQUALITY integerMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE)
attributeTypes: ( 2.16.840.1.113719.1.301.4.51.1
                NAME 'krbExtraData'
                EQUALITY octetStringMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.40)
attributeTypes: ( 2.16.840.1.113719.1.301.4.52.1
                NAME 'krbObjectReferences'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributeTypes: ( 2.16.840.1.113719.1.301.4.53.1
                NAME 'krbPrincContainerRef'
                EQUALITY distinguishedNameMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
objectclasses: ( 2.16.840.1.113719.1.301.6.1.1 NAME 'krbContainer' SUP top STRUCTURAL MUST ( cn ) )
objectclasses: ( 2.16.840.1.113719.1.301.6.2.1 NAME 'krbRealmContainer' SUP top STRUCTURAL MUST
( cn )
                MAY ( krbMKey $ krbUPEnabled $ krbSubTrees $ krbSearchScope $ krbLdapServers $
krbSupportedEncSaltTypes $ krbDefaultEncSaltTypes $ krbTicketPolicyReference $ krbKdcServers $
krbPwdServers $ krbAdmServers $ krbPrincNamingAttr $ krbPwdPolicyReference $ krbPrincContainerRef )
)
objectclasses: ( 2.16.840.1.113719.1.301.6.3.1
                NAME 'krbService'
                SUP top
                ABSTRACT
                MUST ( cn )
                MAY ( krbHostServer $ krbRealmReferences ) )
objectclasses: ( 2.16.840.1.113719.1.301.6.4.1 NAME 'krbKdcService' SUP krbService STRUCTURAL )
objectclasses: ( 2.16.840.1.113719.1.301.6.5.1
                NAME 'krbPwdService' SUP krbService STRUCTURAL )
objectclasses: ( 2.16.840.1.113719.1.301.6.8.1
                NAME 'krbPrincipalAux'
        SUP top
                AUXILIARY
                MAY ( krbPrincipalName $ krbUPEnabled $ krbPrincipalKey $ krbTicketPolicyReference
$ krbPrincipalExpiration $ krbPasswordExpiration $ krbPwdPolicyReference $ krbPrincipalType $
krbPwdHistory $ krbLastPwdChange $ krbPrincipalAliases $ krbLastSuccessfulAuth $ krbLastFailedAuth
$ krbLoginFailedCount $ krbExtraData ) )
objectclasses: ( 2.16.840.1.113719.1.301.6.9.1
                NAME 'krbPrincipal'
                SUP top
                MUST ( krbPrincipalName )
        MAY ( krbObjectReferences ) )
objectclasses: ( 2.16.840.1.113719.1.301.6.11.1
                NAME 'krbPrincRefAux'
                SUP top
                AUXILIARY
                MAY krbPrincipalReferences )
objectclasses: ( 2.16.840.1.113719.1.301.6.13.1
                NAME 'krbAdmService'
                SUP krbService
        STRUCTURAL )
objectclasses: ( 2.16.840.1.113719.1.301.6.14.1
                NAME 'krbPwdPolicy'
                SUP top
                MUST ( cn )
                MAY ( krbMaxPwdLife $ krbMinPwdLife $ krbPwdMinDiffChars $ krbPwdMinLength $
krbPwdHistoryLength ) )
objectclasses: ( 2.16.840.1.113719.1.301.6.16.1
                NAME 'krbTicketPolicyAux'
                SUP top
                AUXILIARY
                MAY ( krbTicketFlags $ krbMaxTicketLife $ krbMaxRenewableAge ) )
objectclasses: ( 2.16.840.1.113719.1.301.6.17.1
                NAME 'krbTicketPolicy'
                SUP top
                MUST ( cn ) )
```

/etc/pam.d/system-auth

```
auth        required      pam_env.so
```

```
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        sufficient    pam_krb5.so use_first_pass debug
auth        sufficient    pam_ldap.so use_first_pass debug
auth        required      pam_deny.so

account     required      pam_unix.so broken_shadow
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
account     [default=bad success=ok user_unknown=ignore] pam_ldap.so debug
account     [default=bad success=ok user_unknown=ignore] pam_krb5.so debug
account     required      pam_permit.so

password    requisite     pam_cracklib.so try_first_pass retry=3
password    sufficient    pam_unix.so md5 shadow nullok try_first_pass use_authtok
password    sufficient    pam_krb5.so use_authtok debug
password    sufficient    pam_ldap.so use_authtok debug
password    required      pam_deny.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session     required      pam_unix.so
session     optional      pam_krb5.so debug
session     optional      pam_ldap.so debug
```

/etc/ldap.conf

```
base dc=example, dc=com
uri ldap://ds.office.example.com
binddn uid=proxyservice, ou=People, dc=example, dc=com
timelimit 120
bind_timelimit 120
idle_timelimit 3600
pam_check_host_attr no
nss_base_passwd ou=People,
nss_base_shadow ou=People,
nss_base_group ou=Groups,
sudoers_base ou=People,
nss_initgroups_ignoreusers
root,ldap,named,avahi,haldaemon,dbus,radvd,tomcat,radiusd,news,mailman,nscd
#ssl start_tls
ssl off
tls_cacertdir /etc/openldap/cacerts
pam_password md5
tls_checkpeer no
use_sasl on
sasl_auth_id proxyservice@example.com
pam_sasl_mech GSSAPI
#pam_check_host_attr yes
pam_filter &(objectclass=posixAccount)(host=ares)
#pam_check_service_attr yes
bind_policy soft
#sudoers_debug 1
```

/etc/krb5.conf

```
[libdefaults]
default_tgs_enctypes = des-cbc-crc des-cbc-md5
default_tkt_enctypes = des-cbc-crc des-cbc-md5
permitted_enctypes = des3-hmac-sha1 des-cbc-crc des-cbc-md5
default_realm = example.com
kdc_timesync = 1
ccache_type = 4
forwardable = true
proxiable = true

[realms]
example.com = {
        admin_server = DS.OFFICE.example.com
        default_domain = OFFICE.example.com
                 kdc = ds.office.example.com
        admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
        acl_file = /var/kerberos/krb5kdc/kadm5.acl
        dict_file = /var/kerberos/krb5kdc/kadm5.dict
        key_stash_file = /var/kerberos/krb5kdc/.k5.OFFICE.example.com
        database_module = openldap_ldapconf
        }
```

```
[domain_realm]
    .office.example.com = example.com
    .example.com = example.com
    .office.aurorisoft.com = example.com
    .aurorisoft.com = example.com

[logging]
kdc = FILE:/var/log/kdc.log
admin_server = FILE:/var/log/kadm.log
kdc = SYSLOG:INFO:DAEMON
admin_server = SYSLOG:INFO:DAEMON
default = SYSLOG

[appdefaults]
# Settings for Red Hat
    pam = {
krb4_convert = false
    }
# Settings for Solaris
    kinit = {
forwardable = true
renewable = true
    }
```

/etc/nsswitch.conf

```
passwd:     files
shadow:     files
group:      files
hosts:      files dns
bootparams: nisplus [NOTFOUND=return] files
ethers:     files
netmasks:   files
networks:   files
protocols:  files
rpc:        files
services:   files
netgroup:   files
publickey:  nisplus
automount:  files
aliases:    files nisplus
```

Return Value of gss_accept_sec_context:

```
GSS_S_BAD_BINDINGS
The input_token parameter contains different channel bindings from those
specified with the input_chan_bindings parameter.

GSS_S_BAD_MECH
The security mechanism used by the context initiator is not available on the
acceptor system.

GSS_S_BAD_SIG
The received input token contains an incorrect signature.

GSS_S_COMPLETE
The routine completed successfully.

GSS_S_CONTINUE_NEEDED
Control information in the returned output token must be sent to the initiator
and a response must be received and passed as the input_token argument to a
continuation call to the gss_accept_sec_context() routine.

GSS_S_CREDENTIALS_EXPIRED
Credentials are no longer valid.

GSS_S_DEFECTIVE_CREDENTIAL
Consistency checks performed on the credential structure referenced by the
verifier_cred_handle parameter failed.
```

```
GSS_S_DEFECTIVE_TOKEN
Consistency checks performed on the input token failed.

GSS_S_DUPLICATE_TOKEN
The token is a duplicate of a token that already has been processed. This is a
fatal error during context establishment.

GSS_S_FAILURE
The routine failed for reasons that are not defined at the GSS level. The
minor_status return parameter contains a mechanism-dependent error code
describing the reason for the failure.

GSS_S_NO_CONTEXT
The context identifier provided by the caller does not refer to a valid
security context.

GSS_S_NO_CRED
No credentials are available or the credentials are valid for context
initiation use only.

GSS_S_OLD_TOKEN
The token is too old to be checked for duplication against previous tokens.
This is a fatal error during context establishment.
```

.htaccess

```
AuthType Kerberos
AuthName "Kerberos Login"
#For the reasons of backwards compatibility the values KerberosV4 and KerberosV5 are also
supported. Their use is not recommended though, for finer setting use following three options.
KrbMethodNegotiate off
#To enable or disable the use of the Negotiate method. You need a special support on the browser
side to support this mechanism.
KrbMethodK5Passwd on
#KrbMethodK4Passwd off
#If set to off this directive allow authentication controls to be pass on to another modules. Use
only if you really know what you are doing.
KrbAuthoritative on
#This option can be used to disable the verification tickets against local keytab to prevent KDC
spoofing atacks. It should be used only for testing purposes. You have been warned.
KrbVerifyKDC off
KrbAuthRealms EXAMPLE.COM
#For specification the service name that will be used by Apache for authentication. Corresponding
key of this name must be stored in the keytab.
KrbServiceName proxyservice
Krb5Keytab /etc/proxyservice.keytab
KrbSaveCredentials off
<Limit GET POST>
    require user crypt@EXAMPLE.COM givi@EXAMPLE.COM
# valid-user — all valid users from kerberos database
#    require valid-user
</Limit>
```