Kerberos и электронная почта.

mkondrin из hppi.troitsk.ru

Аннотация

В данной статье рассмотрен способ интеграции smtp-сервера Postfix и imap/pop3 сервера Cyrus-IMAP в систему единой регистрации пользователей Heimdal-Kerberos. Также показан пример использования этой системы в гетерогенном окружении при помощи почтового клиента Thunderbird.

1 Электронная почта - Postfix и Cyrus-IMAP

В сети существует достаточно много информации об установке и тонкой настройке этих двух программ. Я не ставлю себе целью осветить все детали конфигурации, к примеру, их взаимодействие с программами контроля спама и антивирусами, “виртуализации” пользователей и почтовых ящиков, настройке почтового транспорта и т.д. Ниже будет рассмотрена минимальная конфигурация, которая позволяет аутентифицировать пользователей для работы с электронной почтой с помощью Kerberos, причем чем меньше потребуется изменений в стандартные конфигурационные файлы, прилагаемым к этим программам, тем, с точки зрения данной статьи, лучше.

Итак, напомню задачу. Требуется настроить IMAP/SMTP сервер таким образом, чтобы зарегистрированный в секторе Kerberos пользователь мог получать/отправлять свою почту без дополнительной регистрации в секторе Kerberos, только на основании действительного супербилета, полученного им ранее. Либо, в том случае если почтовый клиент не может обеспечить такой функциональности, пароль, вводимый пользователем, сверялся бы с базой данных паролей, хранимых в Kerberos. Последний вариант обеспечивает совместимость со старыми или недостаточно функциональными почтовыми клиентами.

Аутентификация пользователя для протокола IMAP нужна, хотя бы для того, чтобы каждый пользователь мог читать только свои письма и не имел возможности заглядывать в чужие ящики. В то же время аутентификация для SMTP не столь обязательна и является только средством защиты от несанкционированного релея почты. Протокол SMTP разрешает передачу почты от пользователя находящегося в одном домене через сервер находящийся в другом для получателя в

третьем, чем в недавнем прошлом активно пользовались спаммеры (так называемая ситуация open-relay). Практически все современные SMTP серверы тем или иным способом запрещают такую транспортировку почты, по-умолчанию разрешая передачу сообщений только от отправителя или только получателю, которые находятся в той же сети, что и SMTP-сервер. В принципе, для большинства почтовых сетей такая установка достаточна, и проблемы возникают только тогда, когда требуется обеспечить передачу почты от мобильных пользователей, находящихся вне сети. Запреты/разрешения основанные на адресе отправителя небезопасны сами по себе (помним про ip-spoofing) и, более того, не всегда применимы, поскольку адрес может меняться динамически, а вполне легальный пользователь перемещаться из одной сети в другую. В таком случае идеальным решением было бы дать возможность пользователю аутентифицироваться на SMTP сервере с помощью своего логина и пароля.

В предыдущей части уже была описана процедура настройки взаимодействия библиотеки SASL в сектора Kerberos (с условным названием MYREALM.RU), работающего под управлением сервера Heimdal[1]. Таким образом принцип построения почтовой системы достаточно прост - поскольку у нас уже есть библиотека SASL с поддержкой Kerberos, нужно только встроить SASL в почтовые сервисы, указать используемые этиими сервисами механизмы SASL и зарегистрировать эти сервисы в секторе Kerberos. В дальнейшем нужно будет только выбрать подходящие клиентские программы, которые максимально прозрачно для пользователя будут взаимодействовать с почтовыми сервисами и Kerberos.

Начнем реализацию этой программы с Postfix. Скачиваем последнюю версию с домашней страницы проекта [1] и компилируем:

```
tar xzvf postfix-2.2.8.tar.gz
cd postfix-2.2.
make tidy
make makefiles -i Makefile.in "CCARGS=-DUSE_SASL_AUTH -I/usr/include/sasl
'AUXLIBS=-L/usr/lib -R/usr/lib -lsasl2'
make
make update
./postfix-install -non-interactive
```

Сервер Cyrus-IMAP также разрабатывается в Университете Карнеги-Меллон и доступен по тому же адресу, что и Cyrus-SASL [2]. Несмотря на его название, помимо imap, он поддерживает также протоколы pop3 и nntp. В данной статье, однако,

[1]Напоминаю, что выбор именно этой реализации Kerberos не имеет принципиального значение и обусловлен в основном личными предпочтениями автора в вопросах лицензирования программного обеспечения. Перенос на MIT-Kerberos не должен представлять никаких трудностей.

я ограничусь только протоколом imap. При компиляции Cyrus-IMAP достаточно только указать поддержку SASL:

```
tar xzvf cyrus-imapd-2.3.1.tar.gz
cd cyrus-imapd-2.3.1
./configure \
  --prefix=/usr \
  --sysconfdir=/etc \
  --localstatedir=/var \
  --disable-static \
  --with-cyrus-prefix=/usr/cyrus \
  --with-sasl \
  --enable-gssapi=no
make
make install
```

С помощью опции `--enable-gssapi=no` непосредственная линковка с библиотеками Kerberos отключается. Однако, на конечном результате это не слишком отразится, поскольку эта опция отменяет только сборку сервера kpop – специального протокола доставки почты, разработанного в MIT. Этот протокол существенно отличается от pop3 (не следует думать, что это просто pop3 с поддержкой Kerberos) и мало распространен.

Теперь можно приступать к настройке программ. Прежде всего следует иметь ввиду, что Postfix и Cyrus-IMAP требуют специальных системных пользователей и групп для работы. Хотя они и создаются в процессе установке программ, но стоит лишний раз убедиться, что в `/etc/passwd` и `/etc/group` имеются строки вида:

```
#/etc/passwd
postfix:x:12345:12345:The postfix MTA:/var/spool/postfix:/bin/false
cyrus:x:12365:13:Cyrus Imap Server:/:/bin/false
```

и:

```
#/etc/group
mail::12:mail
postdrop:x:12346:
```

Архитектура Postfix и Cyrus-IMAP в чём-то схожи. В обеих программах центральным звеном является демон, задача которого состоит в прослушивании портов и именованных сокетов. При подключении клиента демон вызывает сервис, который уже осуществляет обмен данными по выбранному протоколу. Центральный демон, который в обеих случаях называется master, по своим функциям напоминает супер-демон inetd, поэтому Postfix и Cyrus-IMAP можно использовать только

как stand-alone сервисы. При установке программ также устанавливаются шаблонные файлы настроек демонов master: для Postfix – /etc/postfix/master.cf, а для Cyrus на выбор предлагается три файла, один из которых нужно скопировать в /etc/cyrus.conf. Для наших целей подойдёт самый простой из них – cyrus-imapd-2.3.1/master/small.conf. Вносить изменения в эти файлы приходится крайне редко, в основном настройка функциональности сервисов производится с помощью редактирования /etc/postfix/main.cf (для Postfix) и /etc/imapd.conf (для Cyrus).

Начнём с настройки Postfix. Лучше всего начать с редактирования файла /etc/postfix/main.cf.default. Изменений немного:

```
queue_directory = /var/spool/postfix
command_directory = /usr/sbin
daemon_directory = /usr/libexec/postfix
sendmail_path = /usr/sbin/sendmail
newaliases_path = /usr/bin/newaliases
mailq_path = /usr/bin/mailq

alias_maps=hash:/etc/postfix/aliases
alias_database=hash:/etc/postfix/aliases
local_recipient_maps=

qmail_owner = postfix
setgid_group = postdrop

myhostname = kdc.myrealm.ru
mydomain = myrealm.ru
myorigin = $mydomain
mydestination = $mydomain, $myhostname, localhost
inet_interfaces = all
mynetworks_style = subnet

mailbox_transport = lmtp:unix:/var/imap/socket/lmtp

smtpd_sasl_auth_enable = yes
smtpd_sasl_local_domain = MYREALM.RU
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions =
        permit_sasl_authenticated,
```

```
        reject_unauth_destination
```

Существенные изменения - имя машины и домена (myhostname и mydomain), и что именно Postfix будет считать локальной сетью (myorigin, mydestination, mynetworks_style). Mailbox_transport определяет способ передачи полученных писем пользователю, в данном случае по протоколу lmtp через именованный сокет, который создаёт Cyrus-IMAP при запуске. С помощью smtpd_sasl_auth_enable включается использование SASL для аутентификации пользователей. Последовательность правил указанная в smtpd_recipient_restrictions позволяет принимать почту без ограничений от авторизованных пользователей (permit_sasl_authenticated) и запрещает релейную пересылку почты незарегистрированным пользователям (reject_unauth_destination). Таким образом для приёма почты на локальные адреса (те что попадают в категорию mydestination) аутентификации не требуется. Параметр же smtpd_sasl_local_domain это просто имя сектора Kerberos.

Но это ещё не всё. Как вы помните, взаимодействие с SASL можно настраивать для каждого приложения индивидуально с помощью конфигурационного файла `/usr/lib/sasl2/<имя приложения>.conf`. Для Postfix имя приложения “smtpd”, поэтому нам нужен файл `/usr/lib/sasl2/smtpd.conf`. Его содержание ничем не отличается от уже созданного ранее файла sample.conf, так что можно просто создать символическую ссылку.

```
#/usr/lib/sasl2/smtpd.conf
pwcheck_method:saslauthd
mech_list:gssapi plain
```

Проверку адресатов получаемых писем мы делать не будем (параметр local_recipient_maps не определён), все полученные письма передаются Cyrus-IMAP без предварительного разбора. Тем не менее нужно создать базу с основными псевдонимами (в основном техническими, например, postmaster) с помощью команды:

```
postalias /etc/postfix/aliases
```

после чего posfix готов к работе:

```
/usr/sbin/postfix start
```

Проверьте по `/var/log/maillog`, что postfix успешно стартовал.

Для Cyrus-IMAP вначале нужно создать его рабочую директорию и дерево каталогов для хранения почтовых ящиков пользователей. Для этого можно воспользоваться скриптом:

```
cd /var
mkdir imap
```

```
for i in proc db log msg socket ptclient ; do
mkdir imap/$i;
done
chown -R cyrus.mail imap
chmod -R 755 imap
cd /var/spool
mkdir imap
chown cyrus.mail imap
chmod 750 imap
```

Обратите внимания на права доступа к созданным директориям. Для передачи почты через сокет `/var/imap/socket/lmtp`, postfix-у, работающему под совсем другим пользователем, чем Cyrus-IMAP, необходимо иметь права на чтение и запуск программ в этой директории.

Настройка Cyrus-IMAP, как уже говорилось, осуществляется с помощью двух конфигурационных файлов. Предполагается, что файл настроек демона master (`/etc/cyrus.conf`) уже скопирован. Для настройки взаимодействия по протоколу imap используется файл `/etc/imapd.conf`:

```
configdirectory: /var/imap
partition-default: /var/spool/imap
admins: cyradmin
altnamespace: yes
loginuseacl: yes
sasl_pwcheck_method: saslauthd
sasl_mech_list: gssapi plain
```

В первых двух строчках определяется рабочие директории Cyrus-IMAP (выше мы их уже создали). Администратором почтовых ящиков назначается cyradmin. Причем в требуется, чтобы принципал cyradmin присутствовал в секторе Kerberos и при этом администратор не должен иметь свой собственный почтовый ящик - это позволяет избежать проблем с его управлением в будущем. Физически почтовый ящик пользователя - это каталог в директории `/var/spool/imap/user`. При отображении в почтовом клиенте структура папок воспроизводит структуру этого каталога, что не всегда удобно. Параметр `altnamespace` приводит почтовый ящик к более привычному “плоскому”виду: с корневой папкой INBOX и всеми подкаталогами приведенными к одному уровню. Параметр `loginuseacl` позволяет с одной стороны создавать общие почтовые ящики, давая административные права на один ящик нескольким пользователям, а с другой позволяет пользователям иметь ”псевдонимы т.е. доступ к ящику с одним именем может быть дан пользователю с совсем иным системным логином. В отличии от Postfix, настройка взаимодействия с SASL

не требует специального файла в каталоге /usr/lib/sasl2, а все параметры SASL определяются непосредственно в файле /etc/imapd.conf. В данном случае настройка аналогична Postfix - добавилась только приставка sasl. После этого можно запустить Cyrus-IMAP:

```
/usr/cyrus/bin/master &
```

и проверить, что в /var/log/syslog не сыпятся сообщения об ошибках.

Перед тем как приступить к созданию почтовых ящиков займёмся заселением базы Kerberos. Как уже говорилось, нам нужно добавить принципала cyradmin и двух принципалов, соответствующих сервисам imap и smtp. Для этого как обычно нужно зарегистрироваться в системе как root, аутентифицироваться в секторе Kerberos как администратор Kerberos и с помощью программы kadmin внести изменения в базу данных Kerberos.

```
#kinit admin/admin
#kadmin
>add cyradmin
>add --random-key imap/kdc.myrealm.ru
>add --random-key smtp/kdc.myrealm.ru
>ext imap/kdc.myrealm.ru
>ext smtp/kdc.myrealm.ru
```

Для сервисов, как обычно, пароли выбираются случайным образом. С помощью команды ext их пароли извлекаются из базы данных Kerberos и записываются в файл /etc/krb5.keytab. Поскольку этот файл содержит пароли в не зашифрованном виде, то права на чтение его следует ограничить. Вместе с тем сервисы, в том числе Postfix и Cyrus-IMAP, работающие под непривилегированными пользователями, должны уметь извлекать свои пароли из этого файла. Для этого самым простым решением будет открыть keytab-файл на чтение выделенной группе (с именем, к примеру, kerberos), в которую нужно будет включить root, postfix и cyrus. Т.е. в файл /etc/group нужно добавить строку вида:

```
kerberos::1000:root,postfix,cyrus
```

а затем поменять владельца и права доступа к keytab-файлу:

```
chown root.kerberos /etc/krb5.keytab
chmod 640 /etc/krb5.keytab
```

Теперь можно приступать к созданию почтовых ящиков для пользователей, а заодно проверить насколько работоспособна аутентификация в Cyrus-IMAP через SASL. Нужно аутентифицироваться с помощью kinit в секторе Kerberos как cyradmin и вызвать утилиту cyradm

```
cyradm kdc.myrealm.ru
kdc.myrealm.ru> cm user.mike
kdc.myrealm.ru> setquota user.mike 10000
```

Таким образом создан ящик ёмкостью 10 МБ для пользователя mike. В этом примере для аутентификации cyradm воспользовался супербилетом принципала cyradmin. Так же можно аутентифицироваться с помощью текстового пароля, но для этого нужно убедиться, что демон saslauthd запущен, и указать в командной строке cyradm имя администратора Cyrus-IMAP и ключ -p для интерактивного ввода пароля:

```
cyradm –u cyradmin –p kdc.myrealm.ru
...
```

После этого наша “игрушечная” почтовая система готова к работе! В итоге команда `ps –aux` должна показать наличие двух процессов master (один для postfix, а другой для cyrus) и ещё несколько процессов saslauthd. После чего можно проверить работоспособность системы с помощью программ smtptest и imtest. Эти программы входят в состав дистрибутива Cyrus-IMAP (каталог imtest). Если они по каким-то причинам не скомпилировались при сборке дистрибутива, то можно сделать это сейчас выполнив make в соответствующем каталоге. Обе программы чем-то напоминают обычный telnet, позволяя подключаться к определённому порту и вводя команды специфические для выбранного протокола в командной строке. Их отличие от telnet только в том, что они выполняют аутентификацию автоматически при запуске, тем самым избавляя вас от необходимости вручную приводить сертификаты к 7-битной форме. Попробуем с их помощью отправить письмо самому себе и затем прочитать его. В начале, разумеется, нужно зарегистрироваться в системе и в Kerberos как обычный пользователь.

```
$kinit mike
$smtptest –m PLAIN –u mike kdc.myrealm.ru
S: 220 kdc.myrealm.ru ESMTP Postfix
C: EHLO example.com
S: 250-kdc.myrealm.ru
S: 250-PIPELINING
S: 250-SIZE 10240000
S: 250-VRFY
S: 250-ETRN
S: 250-AUTH GSSAPI PLAIN
S: 250-AUTH=GSSAPI PLAIN
S: 250 8BITMIME
```

```
Please enter your password:
C: AUTH PLAIN ************
S: 235 Authentication successful
Authenticated.
Security strength factor: 0
MAIL FROM: mike@kdc.myrealm.ru
RCPT TO: mike@kdc.myrealm.ru
DATA
Hello, mike!
.
250 Ok: queued as 9E2495A8
```

Аналогично с помощью imtest прочитаем это “письмо”. В отличии от предыдущего случая, где доступ был получен после предъявления текстового пароля, для получения письма мы используем механизм GSSAPI [2].

```
$imtest -m GSSAPI kdc.myrealm.ru
S: * OK kdc.myrealm.ru Cyrus IMAP4 v2.2.12 server ready
C: C01 CAPABILITY
S: * CAPABILITY IMAP4 IMAP4rev1 ACL QUOTA LITERAL+ MAILBOX-REFERRALS NAME
S: C01 OK Completed
....
S: A01 OK Success (privacy protection)
Authenticated.
Security strength factor: 56
1 select inbox
* FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
* OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen \*)]
* 2 EXISTS
* 1 RECENT
* OK [UNSEEN 2]
* OK [UIDVALIDITY 1137116496]
* OK [UIDNEXT 12]
1 OK [READ-WRITE] Completed
2 fetch 2 body[text]
* 2 FETCH (FLAGS (\Recent \Seen) BODY[TEXT] {14}
Hello, mike!
)
```

[2]Можно было бы использовать и простой текстовый пароль, но для этого программу imtest нужно запускать, как не странно, с ключом -m login.

```
2 OK Completed (0.000 sec)
3 store 1 +flags \deleted
* 1 FETCH (FLAGS (\Deleted \Seen))
3 OK Completed
```

Конечно же регулярно использовать imtest и smtptest для чтения/отправки почты большого удовольствия не доставляет. Поэтому посмотрим какие почтовые клиенты позволяют использовать преимущества аутентификации с помощью GSSAPI в полном объеме.

2 Почтовые клиенты и мобильный доступ

Хотя возможность аутентификации с помощью GSSAPI присутствовала в почтовых службах достаточно давно, но воспользоваться ею было очень сложно из-за практически полного отсутствия подходящих почтовых клиентов. Из консольных клиентов в полном объеме GSSAPI использовал только Pine [3] для аутентификации по протоколам IMAP и SMTP. Популярный клиент Mutt [4] использовал GSSAPI только для протокола IMAP. До последнего времени с графическими почтовыми клиентами дела обстояли еще хуже. Большинство из них, за исключением может быть клиентов от Microsoft традиционно поддерживающими свой протокол NTLM, использовали только вариации открытых текстовых механизмов (PLAIN, LOGIN)[3]. Безопасность соединения обеспечивалась внешними средствами с помощью SSL/TLS. Ситуация изменилась с выходом в январе этого года почтового клиента Thunderbird 1.5, в котором была впервые объявлена поддержка Kerberos. В данной статье я буду рассматривать настройку только этого приложения. Поскольку клиент Thunderbird, как и другие продукты из семейства Mozilla, изначально планировалась как кроссплатформенная программа, то его можно использовать как в Unix-подобных операционных системах так и в системе Windows. В последнем случае, правда, есть несколько небольших ловушек.

Установка и настройка Thunderbird в Linux проблем не вызывает. Нужно только скачать и распаковать бинарную версию Thunderbird с сайта Mozilla [5]. Thunderbird использует GSSAPI непосредственно, не прибегая к услугам SASL. Так что помимо самого почтового клиента вам нужна ещё клиентская часть библиотек Heimdal. Настройка почтовой учетной записи пользователя выполняется как обычно. Нужно только поставить две галочки - одну напротив “Use secure authentication” на закладке Server Settings (для протокола IMAP, меню Edit-> Account Settings и в

[3]Справедливости ради следует упомянуть еще почтовый клиент Mulberry, который поддерживал GSSAPI в полном объеме. К сожалению, этот коммерческий почтовый клиент в основном был распространен на платформе Macintosh и компания продвигавшая его обанкротилась в 2005 году.

дереве слева выбрать Server Settings для нужной учетной записи) и вторую - напротив “Use name and password” в окне свойств Outgoing Server (для протокола SMTP, меню Edit-> Account Settings и в выпадающем окне в дереве слева выбрать Outgoing Server). При наличии супербилета у пользователя учетной записи такая настройка Thunderbird приводит к аутентификации по протоколу GSSAPI, иначе (супербилет пользователем еще не получен) пользователю в доступе будет отказано. Поскольку, как правило, система конфигурируется таким образом, что супербилет запрашивается при входе пользователя в систему, то дополнительного ввода пароля в Thunderbird не требуется. Пользователь сразу же получает доступ к своему почтовому ящику. По логам Cyrus-IMAP и Postfix вы можете убедиться, что при аутентификации действительно используется протокол GSSAPI. Если поле “Use secure authentication” не отмечено, то Thunderbird откатывается к механизму простых текстовых паролей. В этом случае доступ к почтовому ящику возможен только после ввода пароля в выпадающем окошке.

В системе Windows настройка выполняется аналогично. Но недостаток почтового клиента Thunderbird в том, что он не умеет использовать стандартное API Windows для доступа к функциям аутентификации (так называемое SSPI). Поэтому, в настройках по умолчанию Thunderbird в первую очередь нужно запретить использование SSPI - в противном случае это приведёт к некорректному завершению работы программы.

```
user_pref("network.auth.use-sspi", false);
```

Это можно сделать, вручную отредактировав файл prefs.js в каталоге с используемым пользовательским профилем (т.е. `C:\Documents and Settings\[User Name]\Application Data\Thunderbird\Profil`
или введя нужное значение в окне Edit->Preferences->Advanced->Config Editor.

Возможно в следующих версиях Thunderbird будет добавлена недостающая функциональность, но пока у пользователям Windows-версии остается единственная альтернатива – использовать библиотеку Kerberos-for-Windows от MIT. Следует подчеркнуть, что SSPI и библиотека gssapi из пакета KfW используют разные кэши сертификатов. Для пользователя это означает, что даже если супербилет уже получен средствами sspi, то для функционирования KfW (и следовательно Thunderbird) этого недостаточно. Этот сертификат должен быть конвертирован в сертификат MIT (с помощью утилиты ms2mit) или же необходимо получить новый супербилет средствами библиотеки KfW (и графического интерфейса к ней leash32). В статье [6] описана установка пакета KfW и процедура получения обеих супербилетов (MS и MIT) при входе пользователя на свою рабочую станцию.

После того как “правильный” супербилет находится в кэше сертификатов MIT, Thunderbird должна аутентифировать пользователя с помощью механизма GSSAPI.

Однако “должна” не значит, что именно так она и поступит. Если в процессе аутентификации возникает ошибка и Thunderbird настаивает на использовании открытых текстовых паролей, то скорее всего Thunderbird не может найти саму библиотеку GSSAPI. В этом случае для борьбы с этой ошибкой нужно отредактировать еще два параметра по умолчанию Thunderbird (ниже приведены исходные значения):

```
user_pref("network.negotiate-auth.using-native-gsslib", true)
user_pref("network.negotiate-auth.gsslib", "")
```

Первый параметр означает, что Thunderbird использует системную библиотеку gssapi, которая в системе Windows называется gssapi32.dll. Если по каким-то причинам найти её не удаётся, например, потому что она не зарегистрирована в реестре, то нужно поменять значение первого параметра на false, а во втором параметре указать полный путь до соответствующей библиотеки из пакета KfW. Это значение, разумеется, зависит от того, где именно был установлен пакет KfW. У меня, например, этот путь - `C:\KfW\bin\gssapi32.dll`.

Теперь рассмотрим ещё несколько моментов, на которые нужно обратить внимание при переносе “песочницы” в локальную сеть. Во-первых, разумеется, нужно поменять имена принципалов в соответствии с именами компьютеров в локальной сети и именем сектора Kerberos. К примеру, у принципала imap/kdc.myrealm.ru@MYREALM.RU часть имени между / и @ - это имя компьютера, на котором работает сервис, часть имени после @ - название сектора Kerberos - их то вам и нужно привести в соответствие с реальными именами. Во-вторых, ключ сервиса должен присутствовать в keytab-файле на том компьютере, где работает сервис. Точно так же, на том же компьютере должен присутствовать и файл конфигурации SASL (в каталоге `/usr/lib/sasl2`). И последнее - следует задуматься о том, следует ли вам запретить использование открытых текстовых паролей для доступа к сервисам? Если вы все-же решите их оставить, то как тогда обезопасить их передачу? Простые текстовые пароли удобны тем, что они поддерживаются любыми почтовыми клиентами. Однако, чтобы предотвратить перехват простых текстовых паролей при передаче по сети Cyrus-IMAP и Postfix позволяют запретить использование простых текстовых паролей, если канал передачи не зашифрован с помощью SSL/TLS. Для Cyrus-IMAP нужно отредактировать файл `/etc/imapd.conf` добавив параметр:

```
allowplaintext: 0
```

Аналогично можно настроить и Postfix, дописав в main.cf файл строки:

```
smtpd_sasl_security_options=noplaintext,noanonymous
smtpd_sasl_tls_security_options=noanonymous
```

Разумеется, в этом случае нужно дополнительно настроить шифрование канала, но обсуждение этого вопроса выходит за рамки данной статьи.

Если же вы решите текстовые пароли запретить, то следует подумать над тем, каким образом клиенты, находящиеся вне вашего сектора Kerberos могут аутентифицироваться с помощью GSSAPI. Если клиентская часть находится на компьютере, имя и адрес которого постоянны (например, домашний компьютер вашего коллеги), то проще всего включить этот компьютер в ваш сектор Kerberos. Если же на компьютере кроме того, установлен Линукс, то можно также установить на нем локальный сектор Kerberos и наладить траст между этим локальным сектором и корпоративным сектором Kerberos. В любом случае следует убедиться, что доступ к kdc через 88 udp/tcp порт не блокируются брандмауэрами.

Если же доступ к почте осуществляется вполне легальным пользователем, но со "случайного" компьютера и только время от времени, то можно обойтись без включения его в сектор Kerberos. Правда, в этом случае требуется определённые предварительные усилия для того чтобы создать и перенести на этот компьютер билеты Kerberos. Идея метода состоит в том, что раз у пользователя уже есть готовые билеты для получения доступа к сетевым сервисам, то никакого взаимодействия с KDC не требуется и, следовательно, на клиентском компьютере не нужно настраивать сектор Kerberos. Хотя при этом по прежнему в силе остается требование, чтобы на компьютере были доступны библиотеки gssapi и, разумеется, Thunderbird.

Рассмотрим пример - ваш коллега отправляется в командировку, время прибытие на место и отбытия ему известно. Он собирается просматривать свою почту, используя компьютер принимающей организации. Тогда перед отъездом ему нужно получить несколько билетов с вашего KDC, например с помощью такого скрипта:

```
#/bin/bash
for i in $@ ; do
kinit -A --renewable --forwardable --start-time=$i -r 7d -l 1d  -c $HOME/
kgetcred -c $HOME/krb5cc_$i smtp/kdc.myrealm.ru
kgetcred -c $HOME/krb5cc_$i imap/kdc.myrealm.ru
done
```

С помощью этого скрипта пользователь получает безадресный TGT (ключ -A в вызове kinit) действительный в течении суток начиная со дня указанного в качестве входного параметра (`--start-time`). Нужно указать все дни, который ваш сотрудник рассчитывает пробыть в командировке. С помощью полученного TGT генерируются билеты для служб smtp и imap, и весь комплект ключей записываются в файлы вида `krb5cc_<дата>`. Все эти файлы нужно скопировать на какой-нибудь физический носитель (например, флешку), который ваш коллега возьмёт с собой в поездку.

Чтобы воспользоваться билетами, нужно указать Thunderbird, где найти библиотеку gssapi, а библиотеке gssapi сообщить, откуда брать сертификат пользователя. Поскольку все необходимые сертификаты имеются в наличии, то нет необходимости подключаться с удалённого компьютера к контроллеру Kerberos. Если же на флешке вашего коллеги есть еще 45 МБ свободного места, то можно организовать на ней полностью переносимое рабочее окружение из Thunderbird и пакета Kerberos-for-Windows. Таким образом отпадает необходимость в установке и настройке gssapi на клиентском компьютере.

Для этой цели нужно перенести весь каталог с Thunderbird (тот что установлен на `C:\Program Files\...`) на съёмный диск и в него же распаковать пакеты KfW и KfW-extra [7]. В итоге исполняемые файлы thunderbird.exe, leash32.exe и библиотека gssapi32.dll должны оказаться на одном уровне в одном и том же каталоге. При этом следует убедится, что запуск leash32 со съёмного носителя не приводит к сообщениям об ошибке.Затем, в настройках leash32 нужно поменять способ хранения кэша сертификатов с API:krb5cc на FILE:krb5cc. Таким образом скопировав какой-нибудь из полученных ранее сертификатов в файл krb5cc в каталог с leash32 мы делаем этот сертификат доступным библиотеке gssapi. Чтобы не настраивать каждый раз учётную запись пользователя, в тот же каталог стоит также скопировать каталог с пользовательскими настройками (`C:\Documents and Settings\[User Name]\Application Data\Thunderbird\Profil`) для удобства переименовав его, например, в `roaming.def`. Теперь осталось настроить взаимодействие thunderbird с библиотекой gssapi, отредактировав файл `roaming.def\prefs.js`:

```
user_pref("network.auth.use-sspi", false);
user_pref("network.negotiate-auth.using-native-gsslib", false)
user_pref("network.negotiate-auth.gsslib", "gssapi32.dll")
```

Поскольку библиотека gssapi находится в том же каталоге что и исполняемый файл thunderbird.exe, то полный путь указывать не нужно. После чего можно запустить thunderbird с флешки с помощью команды:

```
thunderbird.exe -profile roaming.def
```

и убедиться, что почтовые сервисы доступны и при наличии действительных сертификатов в файле krb5cc от вас не требуется ввод пароля.

На этом мы и завершим настройку почтовой системы. Таким образом нам удалось полностью интегрировать почтовые службы smtp и imap в инфраструктуру Kerberos, а кроссплатформенный почтовый клиент Thunderbird позволяет пользователям Linux и Windows воспользоваться преимуществами single sign-on системы.

Список литературы

[1] SMTP-сервер Postfix.
ftp://ftp.opennet.ru/pub/postfix/official/postfix-2.2.5.tar.gz.

[2] Каталог Cyrus-SASL и Cyrus-IMAP .
ftp://ftp.andrew.cmu.edu/pub/cyrus-mail.

[3] Почтовый клиент Pine.
http://www.washington.edu/pine/getpine/.

[4] Почтовый клиент Mutt.
http://www.mutt.org/.

[5] Почтовый клиент thunderbird.
http://ftp.mozilla.org/pub/mozilla.org/thunderbird/releases/1.5.0.7/source/.

[6] М.Кондрин. Развертываем heimdal kerberos. *Системный Администратор*, 7, 2005.
http://www.samag.ru/cgi-bin/go.pl?q=articles;n=07.2005;a=06.

[7] Версия MIT-Kerberos для MSWindows.
http://web.mit.edu/kerberos/dist/kfw/3.0/kfw-3.0/kfw-3.0.zip,
http://web.mit.edu/kerberos/dist/kfw/3.0/kfw-3.0/kfw-3.0-extra.zip.