Якимчук Сергій

Налаштування поштового сервера на базі Postfix, Dovecot та RoundCube

2016
http://yakim.org.ua

Зміст

Від автора

Шановні читачі! На мою думку на сьогодні дуже замало технічної літератури українською мовою і саме тому я і написав цей матеріал.

Все описане нижче перевірено та працює. Звісно в різних дистрибутивах певні моменти налаштування можуть відрізнятися в залежності від версій програмного забезпечення.

Цей матеріал в електронному вигляді розповсюджується безкоштовно. Єдине прохання — не треба перекладати російською.

Всі зауваження, інформацію про помилки в тексті, а також побажання з приводу майбутніх інструкцій надсилайте на **yakim@yakim.org.ua**.

Якщо у вас виникло бажання віддячити автору — ви можете перерахувати певну суму на ваш власний розсуд на картку ПриватБанка 5168 7554 0234 8029

Вступ

Сьогодні кожна компанія використовує електронну пошту як один з основних засобів комунікацій у бізнесі. Необхідною умовою ефективного застосування пошти є наявність спеціальної програми — поштового сервера. Створювати корпоративні ящики за допомогою публічних сервісів в Інтернеті вже давно вважається свого роду дурним тоном. До того ж, такий підхід абсолютно небезпечний і супроводжується труднощами в керуванні.

Використання власного корпоративного поштового сервера має ряд переваг:

- можливість створення необмеженої кількості поштових доменів (для різних компаній, юридичних осіб, напрямків, підрозділів) на одному сервері;
- можливість створення необмеженої кількості поштових скриньок для різних поштових доменів;
- можливість контролю вхідної та вихідної пошти;
- створення поштових скриньок необмежених розмірів (обмеження обумовлюється тільки технічними характеристиками сервера)
- підтримка протоколів передачі з шифруванням пошти;
- можливість централізованого зберігання пошти на сервері;
- антивірусний захист вхідної/вихідної пошти;
- захист від спаму;
- web-інтерфейс доступу до електронної пошти з шифруванням даних, що передаються.

Поштовий сервер, або сервер електронної пошти — в системі пересилки електронної пошти так зазвичай називають агент пересилки повідомлень.

В Linux-системах сервери доставки повідомлень і сервери передачі повідомлень розділені і адміністратор зв'язує їх між собою самостійно. Найбільш популярна комбінація: Postfix та Dovecot.

Що таке Postfix

Postfix — це агент передачі пошти (Mail Transfer Agent).

Postfix є вільним програмним забезпеченням.

Postfix створювався як альтернатива Sendmail. Вважається, що Postfix швидше працює, легше в адмініструванні, більш захищений і, що важливо, сумісний з Sendmail.

Спочатку Postfix був розроблений Вейтсом Венемой в той час, коли він працював у Дослідницькому центрі імені Томаса Уотсона компанії IBM. Перші версії програми стали доступні в середині 1999 року.

Postfix вирізняється продуманою модульної архітектурою, яка дозволяє створити дуже надійну і швидку поштову систему. Так, наприклад, привілеї root потрібні тільки для відкриття 25 порту, а демони, які виконують основну роботу, можуть працювати непривілейованим користувачем в ізольованому оточенні, що дуже позитивно позначається на безпеці.

Архітектура Postfix виконана в стилі UNIX — де прості програми виконують мінімальний набір функцій, але виконують їх швидко та надійно. При простої поштової системи непотрібні демони можуть припиняти свою роботу, вивільняючи тим самим пам'ять, а при необхідності знову запускаються master-демоном.

Також варто відзначити більш просту і зрозумілу конфігурацію в порівнянні з Sendmail і меншу ресурсоємність, особливо під час простою поштової системи.

Postfix сумісний з BSD, HP-UX, IRIX, GNU/Linux, Mac OS X, Solaris, фактично може бути зібраний на будь якій Unix-подібній операційній системі, що підтримує POSIX і має компілятор C.

Що таке Dovecot

Dovecot — вільний IMAP- і POP3-сервер, що розробляється з розрахунком на безпеку, гнучкість налаштування та швидкодію. Перший реліз відбувся у 2002 році

Dovecot підтримує формати поштових скриньок MBox та Maildir, а також власні формати DBox та Cydir

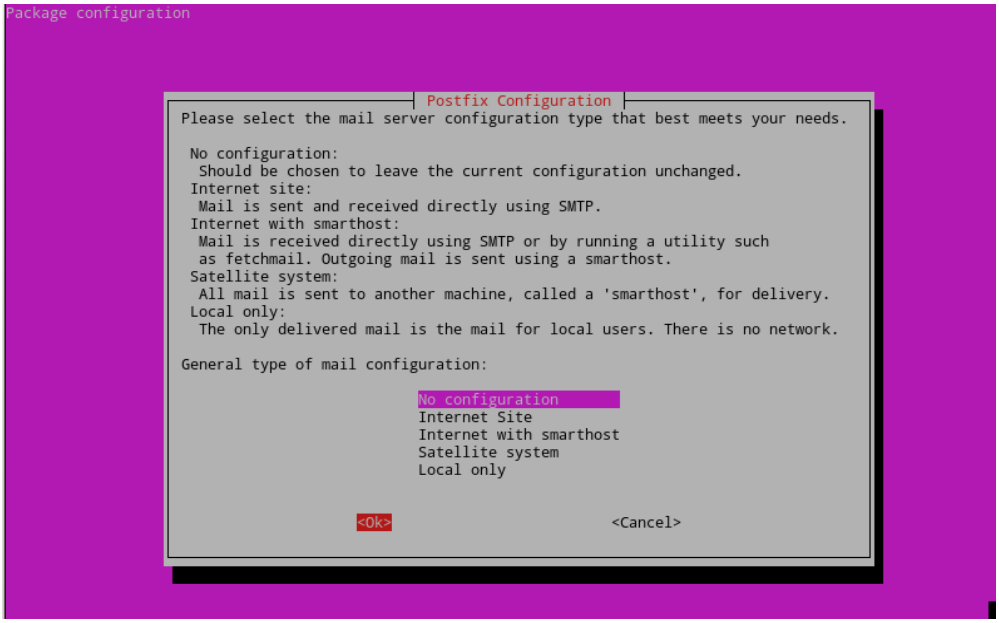
Особливості Dovecot, це:

- Висока швидкодія, завдяки індексації вмісту ящиків.
- Велика кількість підтримуваних механізмів зберігання аутентифікаційної інформації (включаючи LDAP) і самої аутентифікації (підтримується SSL).
- Власна реалізація SASL. Postfix 2.3+ та Exim 4.64+ можуть аутентифікуватися безпосередньо через Dovecot.
- Можливість розширення функціоналу за допомогою плагінів.
- Можливість модифікації індексів з декількох комп'ютерів — що дозволяє йому працювати з NFS та кластерними файловими системами.
- Підтримка різних видів квот
- Підтримка різних ОС: Linux, Solaris, FreeBSD, OpenBSD, NetBSD і Mac OS X
- Простота налаштування.
- Суворе дотримання стандартів — Dovecot один з небагатьох хто проходить тест на відповідність усім стандартам IMAP. До речі, прошу звернути увагу, що Microsoft Exchange цей тест не проходить.

Встановлення поштового сервера

Спочатку встановлюємо Postfix та Dovecot:
$sudo aptitude install postfix dovecot-core dovecot-imapd

Відмовляємося від запропонованих варіантів налаштувань. Мається на увазі, що все налаштуємо самі без жодного автоматизму з боку розробників.



Постінсталляційний скрипт повідомить, що в такому стані Postfix працювати не може.
Створимо відсутній файл конфігурації:
$ sudo touch /etc/postfix/main.cf

На цьому встановлення завершене.

Підготовка до налаштування поштового сервера:

Створимо місце для зберігання пошти на сервері для нашого поштового домену study.local:
#mkdir -p /var/spool/mail/study.local

Створимо групу *virtual* та користувача *virtual*:
#groupadd -g 5000 virtual
#useradd -g virtual -u 5000 virtual

Для них ми призначили uid та gid 5000. Число було обрано довільно, як достатньо велике.

Вкажемо власника та права доступу до теки з поштою:
#chown virtual:virtual /var/spool/mail/study.local
#chmod 770 /var/spool/mail/study.local

Налаштування Postfix

Відкриваємо на редагування файл */etc/postfix/main.cf* та приведемо його до наступного вигляду:

#Так наш сервер буде представлятися при відправці та отриманні пошти
smtpd_banner = $myhostname ESMTP (ubuntu)

```
biff = no #Вимикаємо використання comsat
#Забороняємо автоматично доповнювати неповне доменне ім'я в адресі листа
append_dot_mydomain = no
queue_directory = /var/spool/postfix #Вказуємо теку черги для Postfix
myhostname = mail.study.local #Вказуємо ім'я нашого хоста
alias_maps =
myorigin = study.local
mydestination = localhost #Вказуємо, для яких доменів будемо приймати пошту

#Вказуємо, для яких віртуальних доменів будемо приймати пошту
virtual_mailbox_domains = study.local
virtual_mailbox_base = /var/spool/mail/ #Початок шляху для зберігання пошти
virtual_alias_maps = hash:/etc/postfix/virtual #Файл з описом поштових аліасів
virtual_mailbox_maps = hash:/etc/postfix/vmailbox #Файл з описом поштових скриньок
virtual_minimum_uid = 100
virtual_uid_maps = static:5000
virtual_gid_maps = static:5000

mynetworks = 127.0.0.0/8 #Вказуємо список довірених підмереж
inet_interfaces = all #Приймаємо з'єднання на всіх інтерфейсах

#Описуємо авторизацію через Dovecot
smtpd_sasl_auth_enable = yes
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_helo_required = yes #Обов'язково при з'єднанні вимагати helo
#Далі налаштовуємо фільтри прийому/відправлення пошти

#Правила, що діють на етапі команди HELO
smtpd_helo_restrictions = permit_mynetworks,
                          permit_sasl_authenticated,
                          reject_unknown_client,
                          eject_non_fqdn_hostname,
                          reject_invalid_hostname,
                          reject_unknown_hostname

#Правила, що діють на етапі команди rcpt to
smtpd_recipient_restrictions = permit_mynetworks,
                          permit_sasl_authenticated,
                          reject_unauth_destination,
                          reject_unknown_sender_domain,
                          reject_unknown_recipient_domain,
                          reject_non_fqdn_recipient,
                          reject_non_fqdn_sender
```

Правила фільтрації поштових повідомлень.

Якщо наш сервер буде приймати та пересилати будь-які поштові повідомлення, то дуже швидко він буде внесений до всіх чорних списків, як розповсюджувач спаму. Крім того значно підвищиться навантаження на наш сервер — всі вхідні листи треба обробляти не лише самому поштовому серверу, а ще й антивірусу та антиспаму. Для зниження навантаження на сервер, а також для блокування небажаних повідомлень ще на етапі їх прийняття, використовуються правила фільтрації вхідних повідомлень. Вони записані в нашому файлі конфігурації в блоках *smtpd_helo_restrictions* та *smtpd_recipient_restrictions*. У нас ці правила виглядають наступним чином:

smtpd_helo_restrictions = permit_mynetworks,
permit_sasl_authenticated,
reject_unknown_client,
eject_non_fqdn_hostname,
reject_invalid_hostname,

smtpd_recipient_restrictions = permit_mynetworks,
permit_sasl_authenticated,
reject_unauth_destination,
reject_unknown_sender_domain,
reject_unknown_recipient_domain,
reject_non_fqdn_recipient,
reject_non_fqdn_sender

Роздивимось їх докладніше:
permit_mynetworks — приймати всі листи з довіреної зони
permit_sasl_authenticated — приймати всі листи по з'єднанням з авторизацією
reject_unauth_destination — відкидати листи, які не відносяться до доменів, що ми обслуговуємо
reject_unknown_sender_domain — відкидати листи від невідомого домену відправника
reject_unknown_recipient_domain — відкидати листи для невідомого домену отримувача
reject_non_fqdn_recipient — відкидати листи для неповного домену отримувача
reject_non_fqdn_sender — відкидати листи від неповного домену відправника
reject_non_fqdn_hostname — відкидати листи, якщо ім'я сервера відправника неповне
reject_invalid_hostname — відкидати листи, якщо неправильне ім'я сервера відправника
reject_unknown_hostname — відкидати листи, якщо невідоме ім'я сервера відправника

Ці правила виконуються по черзі від першого до останнього. Якщо лист не потрапив під жодне правило, то його буде прийнято.

Додаткові правила фільтрації поштових повідомлень.

Для зменшення кількості спаму має сенс додати ще декілька правил фільтрації

В кінець блоку *smtpd_helo_restrictions* допишемо
check_helo_access hash:/etc/postfix/helo.list

А в блок *smtpd_recipient_restrictions* після правила *permit_sasl_authenticated* вставимо

check_sender_access hash:/etc/postfix/ext_sender,

Створимо файл */etc/postfix/helo.list*
#touch /etc/postfix/helo.list

Відкриємо його на редагування та запишемо в нього рядок:
mail.study.local 550 Don't use my hostname

Та створимо з нього індексовану мапу:
#postmap /etc/postfix/helo.list

Створимо файл */etc/postfix/ext_sende*r
#touch /etc/postfix/ext_sender

Відкриємо його на редагування та запишемо в нього рядок:
study.local 550 Do not use my domain in your envelope sender

Та створимо з нього індексовану мапу:
#postmap /etc/postfix/ext_sender

Правилом *check_helo_access* буде здійснюватись перевірка того, що нам надішле сервер-відправник в команді HELO. В тому випадку, якщо він представиться нашим же іменем (рядок *mail.study.local* в файлі */etc/postfix/helo.list*), то з'єднання буде розірвано. В нормальній ситуації жоден сервер не може представлятися нашим іменем, тому, скоріш за все, це буде сервер, що розповсюджує спам.

Правилом *check_sender_access* буде здійснюватись перевірка адреси відправника листа. В тому випадку, коли відправник знаходиться в нашому ж домені (рядок *study.local* у файлі */etc/postfix/ext_sender*), буде відмовлено в прийомі такого листа. З точки зору нашого сервера лист з таким відправником не може прийти від якогось відправника ззовні. Такий лист може бути відправлений лише користувачем нашого сервера, але тоді з'єднання буде захищене логіном та паролем і в цьому випадку лист буде прийнятий за правилом *permit_sasl_authenticated*, яке стоїть попереду і тому спрацює раніше.

Також має сенс додати ще одну перевірку:
reject_unknown_client
Це правило забороняє приймання листа в тому випадку, коли у клієнта неправильні налаштування DNS — відсутнє або неправильне доменне ім'я (DNS запис типу A) або відсутня або неправильна зворотня зона (DNS запис типу PTR). Вставляти це правило треба в блок *smtpd_helo_restrictions* зразу після *permit_sasl_authenticated*

Використання чорних списків розповсюджувачів спаму - DNSBL

DNSBL — DNS blacklist або DNS blocklist — списки хостів, збережені з використанням системи архітектури DNS. Зазвичай використовуються для боротьби зі спамом. Поштовий сервер звертається до DNSBL і перевіряє в ньому наявність IP-адреси клієнта, з якого він приймає повідомлення. При позитивній відповіді вважається, що відбувається спроба

прийому спам-повідомлення. Серверу відправника повідомляється помилка 5xx (невиправна помилка) і повідомлення не приймається. Поштовий сервер відправника створює «відмовну квитанцію» відправнику про недоставку пошти.

Для налаштування використання чорних списків серверів потрібно в кінець блоку правил *smtpd_helo_restrictions* додати рядки типу

reject_rbl_client sbl.spamhaus.org

Де sbl.spamhaus.org це і є адреса сервіса DNSBL

Таких списків в інтернеті знайти можна дуже багато, але найчастіше використовуються наступні:

sbl.spamhaus.org
cbl.abuseat.org
dnsbl.sorbs.net

При використанні DNSBL слід бути дуже уважним. Безкоштовні сервіси можуть в будь-яку мить припинити роботу, внести в списки нормальні сервери або створити інші неприємності. Тому користуватися ними слід дуже обережно.

Налаштування Dovecot

Проведемо налаштування Dovecot версії 2.xx.

В теці */etc/dovecot*, на відміну від старих версій програми, ми маємо багато файлів конфігурації. При чому навіть з підтеками.

Звичайно можна всю конфігурацію звести в один файл, але це не дуже правильно, бо суперечить тому, що задумали розробники.

Відкриємо основний файл конфігурації */etc/dovecot/dovecot.conf* та приведемо його до наступного вигляду:

```
# За яким протоколом працюємо
protocols = imap
# Слухаємо з'єднання на всіх інтерфейсах по протоколу IPv4
listen = *
# Робоча тека
base_dir = /var/run/dovecot/
# Ім'я інстансу (для відображення в лозі)
instance_name = dovecot
# Рядок привітання
login_greeting = Dovecot ready.
# Відключати клієнтські з'єднання при виключенні або перезавантаженні майстер-сервісу
shutdown_clients = yes
# Сокет керуючого сервісу doveadm
doveadm_socket_path = doveadm-server
# Підключаємо окремі файли конфігурації
!include conf.d/*.conf
```

Тепер переходимо до теки */etc/dovecot/conf.d*
Відкриємо в ній файл 10-auth.conf і пропишемо в ньому два рядки:

disable_plaintext_auth = no
auth_mechanisms = plain login

а також в кінці цього файлу закоментуємо рядок
!*include auth-system.conf.ext*

та розкоментуємо
!include auth-passwdfile.conf.ext

Далі відредагуємо файл 10-mail.conf
mail_location = maildir:/var/spool/mail/%d/%n
mail_uid = 5000
mail_gid = 5000
mail_privileged_group = virtual
valid_chroot_dirs = /var/spool/mail/

Далі нас буде цікавити файл *10-master.conf*

service imap-login {
inet_listener imap {
#port = 143
}
inet_listener imaps {
#port = 993
#ssl = yes
}
}
service auth {
 # Postfix smtp-auth
unix_listener /var/spool/postfix/private/auth {
mode = 0666
}
 # Auth process is run as this user.
user = postfix
group = postfix
}
І, нарешті, в файлі *10-ssl.conf* треба прописати
ssl = no

Наостаннє, треба видалити файл *15-mailboxes.conf*

Створення поштових скриньок та псевдонімів

Тепер створимо користувача та поштову скриньку для нього:
Логін — user@study.local
Пароль — password
Адреса — user@study.local

Створимо необхідні файли в Postfix:
 # touch /etc/postfix/vmailbox
 # touch /etc/postfix/virtual

Пропишемо в Postfix дані про нову поштову скриньку. Для цього в файл */etc/postfix/vmailbox* допишемо рядок:
user*@study.local study.local/user/*

Створимо для прикладу аліас на цю поштову скриньку. Для цього в файл */etc/postfix/virtual* допишемо рядок:
postmaster@study.local user@study.local

Та створимо індексовану мапу з цих файлів:
#postmap /etc/postfix/virtual
#postmap /etc/postfix/vmailbox

Тепер потрібно перезапустити Postfix:
 # service postfix restart

Внесемо дані про нашого користувача в Dovecot.
Якщо подивитися файл *auth-passwdfile.conf.ext* то ми побачимо, що логіни та паролі користувачів мають зберігатися у файлі */etc/dovecot/users* зі схемою шифрування CRYPT.
Створимо запис для користувача *user@study.local* з паролем user.
$doveadm pw -s CRYPT -u user@study.local -p password

Отримані дані внесемо до файлу */etc/dovecot/users*
user@study.local:{CRYPT}CaKFEZXiRl/aE:5000:5000

Поштовий веб-інтерфейс RoundCube

RoundCube Webmail — це клієнт для роботи з електронною поштою з веб-інтерфейсом, написаний на PHP з використанням CSS та XHTML і технології AJAX. RoundCube Webmail встановлюється практично на будь-який сервер з підтримкою PHP та MySQL і надає можливість роботи з поштовими скриньками за протоколами IMAP та SMTP.

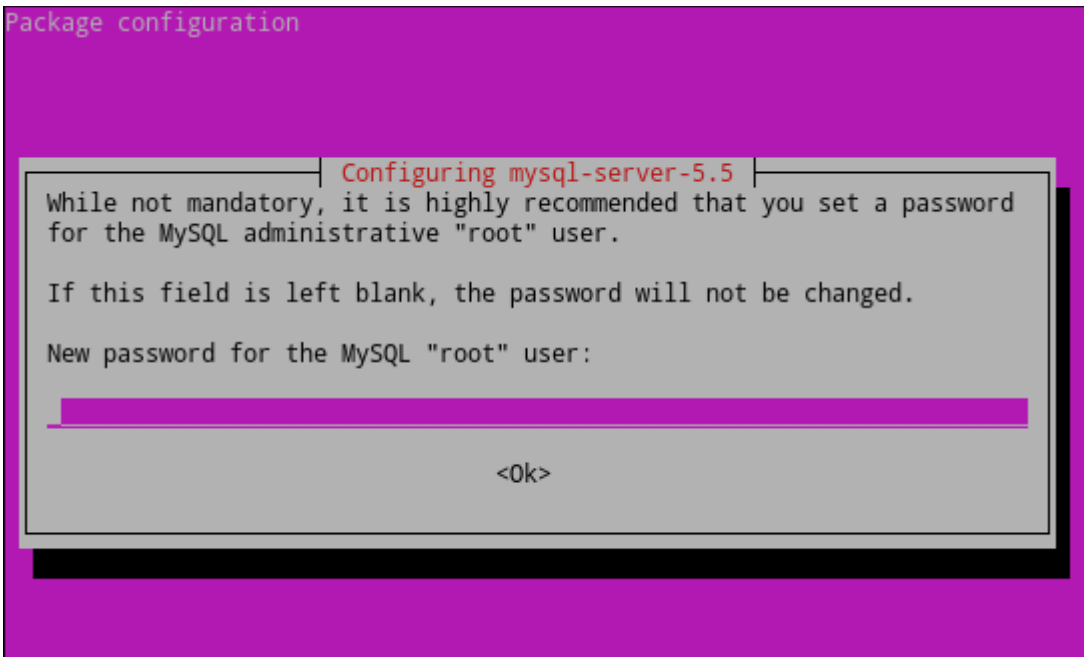
Проект був заснований 18 травня 2005 року. Тоді RoundCube Webmail був скромний клієнт для роботи з електронною поштою. Зараз це потужний поштовий додаток, який майже нічим не поступається звичайним поштовим клієнтам.

RoundCube Webmail доступний за ліцензією GPL та є вільним програмним забезпеченням.

Встановлення roundcube

Для своєї роботи, а точніше для зберігання даних, кешування пошти та внутрішньої адресної книги RoundCube вимагає наявності MySQL-сервера. Так як у нас його в мережі поки-то немає, то встановимо його локально

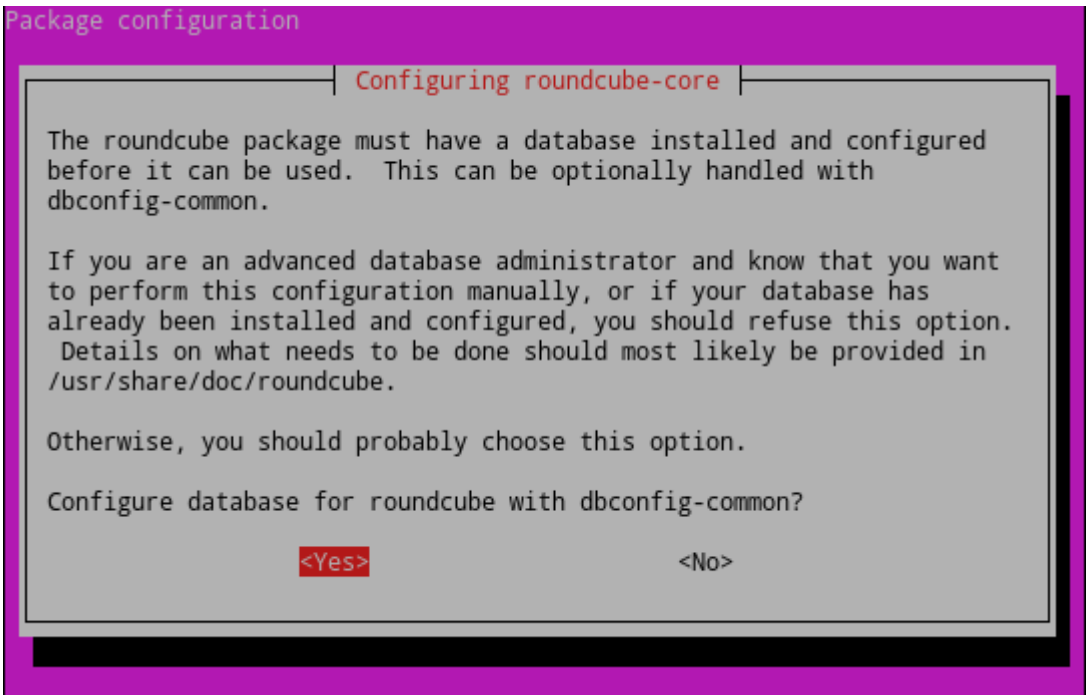
 # apt-get install mysql-server mysql-client



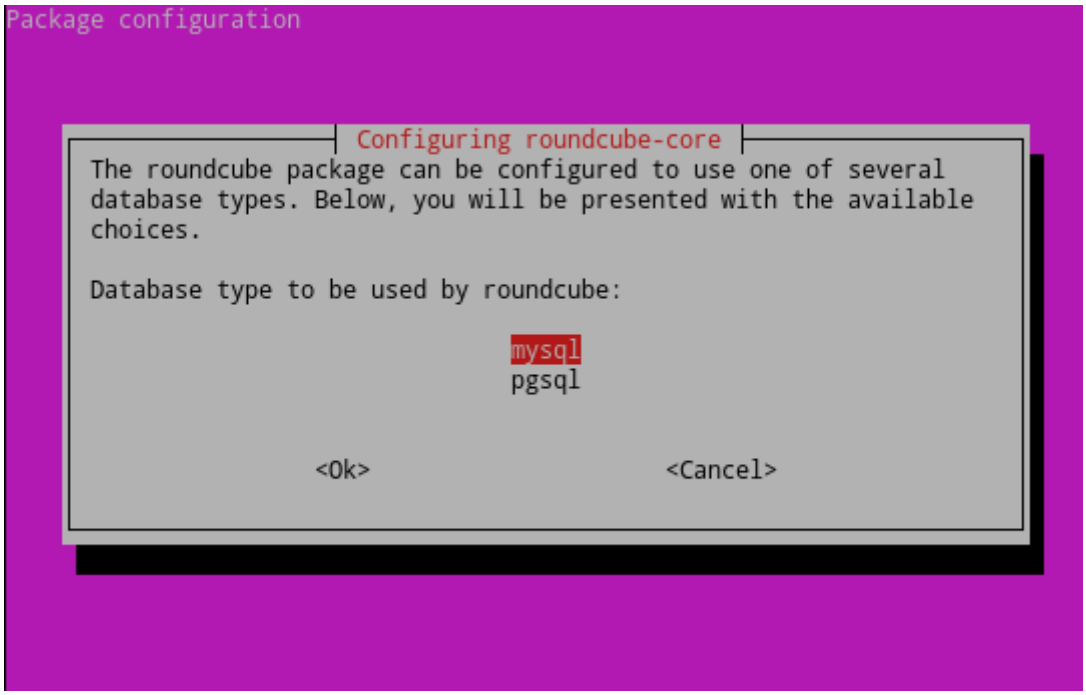
В процесі встановлення введемо пароль для користувача root нашого MySQL-сервера

Тепер можна встановлювати безпосередньо і сам RoundCube
 # apt-get install roundcube

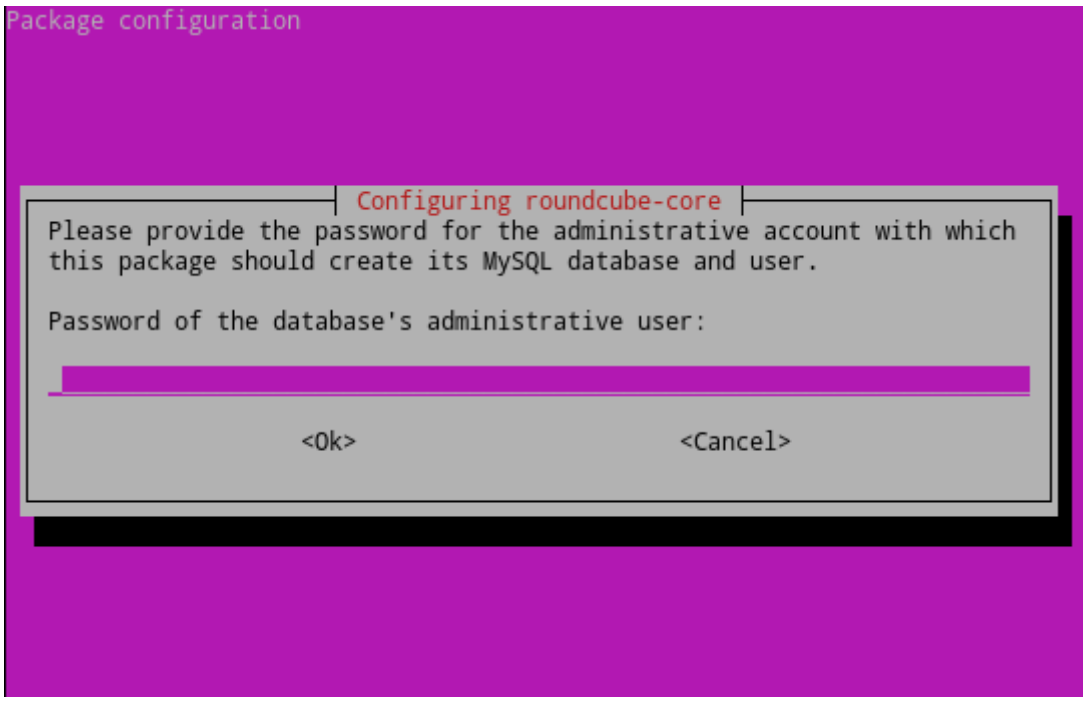
В процесі встановлення нам буде запропоновано автоматично налаштувати базу даних для RoundCube за допомогою dbconfig-common



Оберемо тип бази даних - mysql



Та вводимо пароль користувача root сервера MySQL (в деяких дистрибутивах цей крок може бути пропущений)



Потім нам запропонують ввести пароль для користувача roundcube. Його можна залишити порожнім. В цьому випадку пароль буде створений випадковим чином.

На цьому встановлення завершене.

Зараз необхідно налаштувати RoundCube для коректної роботи з нашим поштовим сервером.

Налаштування RoundCube

Відкриємо файл конфігурації RoundCube — */etc/roundcube/config.inc.php*
В ньому необхідно змінити кілька рядків:

$config['default_host'] = array("127.0.0.1");

$config['smtp_server'] = '127.0.0.1';

Та в файлі */etc/roundcube/defaults.inc.php*
$config['mail_domain'] = 'study.local';

Налаштування сервера Apache

Налаштуємо веб-сервер, для того, щоб поштовий інтерфейс працював по захищеному протоколу https.

Для цього підключимо до Apache необхідні розширення

*# a2enmod ssl**

Тепер підключимо https-сайт за замовчанням
 # a2ensite default-ssl
Та ще в файлі */etc/apache2/conf.d/roundcube* раскоментуємо рядок
Alias /roundcube /var/lib/roundcube

Після цих дій необхідно перезапустити веб-сервер
#service apache2 restart

Також необхідно переконатися в тому, що пакет php5-mcrypt у нас встановлений і після цього виконати команди:
 # ln -s /etc/php5/mods-available/mcrypt.ini /etc/php5/apache2/conf.d/20-mcrypt.ini
 # ln -s /etc/php5/mods-available/mcrypt.ini /etc/php5/cli/conf.d/20-mcrypt.ini

Після цих дій необхідно перезапустити веб-сервер
#service apache2 restart

Тепер веб-інтерфейс для нашого поштового сервера доступний за посиланням https://ip-addr/roundcube

На цьому базове налаштування поштового сервера закінчене.

Налаштування автоматичного створення прихованих копій повідомлень.

Іноді буває необхідність створювати копії листів та відсилати їх на іншу адресу. Для цього у Postfix є механізм створення прихованих копій (bcc – blind carbon copy)

Робити такі копії можна на основі адреси або відправника, або отримувача. Для цього в файлі main.cf можна використовувати параметри sender_bcc_maps або recipient_bcc_maps.

Формат цих рядків буде таким
sender_bcc_maps = type:table
recipient_bcc_maps = type:table

Наприклад нам потрібно перенаправляти всі листи від user@study.local та до user@study.local на адресу chief@study.local.

Створимо файли *bcc_recipient* та *bcc_sender* та запишемо в них однаковий рядок
user@study.local chief@study.local

Створимо індексовані мапи файлів
postmap bcc_recipient
postmap bcc_sender

додамо в файл *main.cf* два рядки
sender_bcc_maps = hash:/etc/postfix/bcc_sender
recipient_bcc_maps = hash:/etc/postfix/bcc_recipient

та перезапустимо сам postfix
service postfix restart

Після цього всі листи від та до user@study.local будуть копіюватися на адресу chief@study.local

Також буває необхідність робити копії всіх листів не від певної адреси, а від певного домену відправника. Для цього треба використовувати інший тип таблиць — regexp.

Зробимо копію всіх листів, які відправляються з домену test.local. Відсилати копії будемо на ту ж саму адресу — chief@study.local. Для цього створимо файл *reg_bcc_sender* та запишемо в нього
/test\.local/ chief@study.local

Один раз ми вже описали параметр sender_bcc_maps і другий раз його описувати вже не потрібно. Просто доповнимо його опис. Приведемо рядок до вигляду
sender_bcc_maps = hash:/etc/postfix/bcc_sender regexp:/etc/postfix/reg_bcc_sender
та перезапустимо сам postfix
service postfix restart

Тепер крім попередніх налаштувань прихованих копій, також будуть копіюватися всі листи з будь-якої адреси домену test.local

Маршрутизація поштового трафіка. Авторизація на сервері вищого рівня.

Зазвичай налаштування маршрутизації поштового трафіка робити немає сенсу. Всі повідомлення будуть маршрутизуватись автоматично, згідно MX записам в DNS домену отримувача. Однак бувають і виключення з цього правила. За допомогою параметра transport_maps можна вказати в явному вигляді сервер, на який буде відправлена пошта для певного домену.

Створимо файл */etc/postfix/transport* і запишемо в нього
test.local smtp:192.168.0.2

Створимо індексовану мапу файла
postmap /etc/postfix/transport

Далі в файл *main.cf* додамо рядок
transport_maps = hash:/etc/postfix/transport

та перезапустимо сам postfix
service postfix restart

Тепер всі листи для користувачів домену test.local будуть відправлені на сервер з адресою 192.168.0.2 і при цьому DNS-запитів поштовий сервер робити не буде.

Інший варіант налаштування маршрутизації — це коли пошту з певних причин не можна відправляти безпосередньо на сервер отримувача і треба використовувати сервер-відправник вищого рівня, тобто зовнішній SMTP-сервер (так званий поштовий релей).

Це налаштування можна зробити за допомогою параметра relayhost. Припустимо, що відправляти пошту нам потрібно через сервер з іменем *mail.test.local* і через порт 587 (SMTP submission port). Для цього в файл main.cf допишемо рядок

relayhost = [mail.test.local]:587

Після перезапуска postfix всі листи будуть відправлені не на сервер отримувача, а на *mail.test.local*.

На цьому етапі може виникнути ще одна проблема. Це трапиться в тому випадку, коли релей-сервер вимагає авторизації. Для того, щоб налаштувати передачу логіна-пароля треба виконати наступні дії

До файла main.cf додаємо рядки

 # Вмикаємо аутентифікацію
smtp_sasl_auth_enable = yes
 # Вказуємо файл з логінами та паролями
smtp_sasl_password_maps=hash:/etc/postfix/sasl_passwd
#Забороняємо анонімну аутентифікацію
smtp_sasl_security_options = noanonymous

Створюємо файл */etc/postfix/sasl_passw*d и записуємо в нього
mail.test.local user@test.local:password
де
mail.test.local сервер, на якому будемо авторизуватися
user@test.local:password логін:пароль

Далі виконуємо команду:

postmap /etc/postfix/sasl_passwd

та перезапускаємо postfix
service postfix restart

Альтернативні методи збереження налаштувань користувачів

Використання MySQL

Облікові дані користувачів можна зберігати також у базі даних. Для цього підходить PostgreSQL та Mysql.

Налаштування в Mysql

Розглянемо використання MySQL, як більш популярний варіант
Зайдемо в MySQL з правами root
$ mysql -uroot -p

Створимо базу даних з іменем maildata для поштової системи
mysql> Create DATABASE maildata;

Створимо користувача mail з паролем mail та надамо йому повний доступ до нашої бази
mysql> GRANT all ON maildata. TO 'mail'@'localhost' IDENTIFIED BY 'mail';*

Перейдемо до щойно створеної бази
mysql> use maildata;

та створимо таблицю користувачів, де будуть зберігатися email-адреса (вона ж логін користувача), пароль цього користувача та закінчення шляху до його поштової теки.
mysql> CREATE TABLE users (email varchar(80) NOT NULL, password varchar(20) NOT NULL, path varchar(80) NOT NULL, PRIMARY KEY(email));

Також створимо таблицю псевдонімів
mysql> CREATE TABLE aliace (source varchar(80) NOT NULL, destination TEXT NOT NULL, PRIMARY KEY(source));

Значення полів аналогічні тому, як ми зберігали їх у файлах.
Тепер заповнимо ці таблиці тестовими даними

mysql> insert into users (email,password,path) values ('admin@study.local','admin','study.local/admin/');
mysql> insert into aliace (source,destination) values ('postmaster@study .local','admin@study .local');

Всі попередні налаштування не обов'язково робити з командного рядка. Все це можна виконати за допомогою веб-інтерфейсу, наприклад phpMyAdmin.

Налаштування Postfix

Встановимо підтримку MySQL Postfix'ом
 # apt-get install postfix-mysql

Додамо в файл */etc/postfix/main.cf* підтримку даних в базі MySQL

virtual_alias_maps = hash:/etc/postfix/virtual mysql:/etc/postfix/aliace.cf
virtual_mailbox_maps = hash:/etc/postfix/vmailbox mysql:/etc/postfix/box-sql.cf

Зверніть увагу на те, що не обов'язково відключати попередні файли даних. Postfix може одночасно використовувати декілька джерел даних.

Створимо файл обробки псевдонімів */etc/postfix/aliace.cf* і запишемо в нього
user = mail # Логін
password = mail # Пароль
dbname = maildata # Назва БД
hosts = 127.0.0.1 # Хост з БД
query = SELECT destination FROM aliace WHERE source = '%s' # SQL запит для обробки псевдонімів

Також створимо файл */etc/postfix/box-sql.cf* для обробки адрес отримувачів

user = mail # Логін
password = mail # Пароль
dbname = maildata # Назва БД
hosts = 127.0.0.1 # Хост з БД
query = SELECT path FROM users WHERE email = '%s' # SQL запит для обробки адрес поштових скриньок

Налаштування Dovecot

Для того, щоб Dovecot міг звертатися за даними користувачів до MySQL, треба встоновити додатковий пакет
 # apt-get install dovecot-mysql

Розкоментуємо в файлі */etc/dovecot/conf.d/10-auth.conf* рядок
!include auth-sql.conf.ext
Зверніть увагу — dovecot, як і postfix може одночасно використовувати декілька баз користувачів.

Далі у файл */etc/dovecot/dovecot-sql.conf.ext* треба внести рядки

driver = mysql
 # параметри з'єднання з сервером БД
connect = host=localhost dbname= maildata user=mail password=mail
default_pass_scheme = PLAIN #паролі зберігаємо у відкритому вигляді
 # рядки SQL-запитів
password_query = SELECT email AS email,password FROM users WHERE email = '%u'
user_query = SELECT email AS user, '5000' AS uid, '5000' AS gid FROM users WHERE email = '%u'

Тепер можна використовувати базу MySQL як сховище налаштувань користувачів та поштових псевдонімів.

Використання LDAP

Також можна під'єднати поштову систему до LDAP і брати всі налаштування звідти. В якості LDAP-сервера можна використовувати контроллер Windows-домена. Це може бути або OpenLDAP, або Windows AD.

Налаштування Postfix

Спочатку встановимо підтримку LDAP в Postfix
 # apt-get install postfix-ldap

Додамо в файл */etc/postfix/main.cf* підтримку даних в базі LDAP

virtual_alias_maps = hash:/etc/postfix/virtual mysql:/etc/postfix/aliace.cf ldap:/etc/postfix/ldapalias
virtual_mailbox_maps = hash:/etc/postfix/vmailbox mysql:/etc/postfix/box-sql.cf ldap:/etc/postfix/ldap_virtual_mailbox_maps.cf
Знову ж таки, ми не відключаємо попередні налаштування, а додаємо нові джерела даних.

Далі потрібно створити файли запитів до LDAP. Запити можуть відрізнятися в залежності від того, яка саме структура LDAP використовується. Надалі буде розглядатися варіант Windows AD. Припустимо, що в домені study.local є спеціальний користувач mailadmin з паролем mailadmin, який має права на читання в дереві LDAP. Контроллер домена має адресу 192.168.0.10. У користувачів домена має бути заповненим атрибут mail, в який вноситься поштова адреса даного користувача. Для роботи з поштовими псевдонімами створюються доменні групи, у яких так само заповнюється атрибут mail. Листи, що приходять на адресу таких груп мають бути передані всім користувачам, що входять до них.

Для обробки поштових адрес користувачів створимо файл */etc/postfix/ldap_virtual_mailbox_maps.cf* і запишемо в нього

server_host = 192.168.0.10
bind = yes
bind_dn = cn=mailadmin,cn=Users,dc=study,dc=local
bind_pw = mailadmin
search_base = cn=Users,dc=study,dc=local
query_filter = (&(mail=%s))
result_attribute = mail
result_format = %d/%u/

Для обробки поштових псевдонімів створимо файл */etc/postfix/ldapalias* і запишемо в нього

server_host = 192.168.0.10
bind = yes

bind_dn = cn=mailadmin,cn=Users,dc=study,dc=local
bind_pw = mailadmin
search_base = cn=Users,dc=study,dc=local
query_filter = (&(objectClass=group)(mail=%s))
leaf_result_attribute = mail
special_result_attribute = member

Тепер Postfix може приймати пошту, вичитуючи дані з домену.

Налаштування Dovecot

Спочатку встановимо підтримку LDAP в Dovecot
 # apt-get install dovecot-ldap

Розкоментуємо в файлі /*etc/dovecot/conf.d/10-auth.conf* рядок
!include auth-ldap.conf.ext

І знову ми не відключаємо старі джерела даних про користувачів, а лише додаємо нове.
В файл /etc/dovecot/dovecot-ldap.conf.ext вносимо інформацію про з'єднання з контроллером домену та про запити щодо користувачів.

hosts = 192.168.0.10
auth_bind = yes
ldap_version = 3
base = cn=Users,dc=study,dc=local
dn = cn=mailadmin,cn=Users,dc=study,dc=local
dnpass = mailadmin
deref = never
scope = subtree
user_filter = (&(ObjectClass=person)(sAMAccountName=%u))
pass_filter = (&(ObjectClass=person)(sAMAccountName=%u))

Зверніть увагу ще на такий момент — в LDAP паролі зберігаються у вигляді хеша. У зв'язку з тим, що хешування це операція не обернена, то перевести хеш одного типу в інший неможливо. Саме тому при налаштуванні поштового сервера ми використовували PLAINTEXT аутентифікацію. Для безпечної передачі логінів та паролей треба використовувати шифровані з'єднання клієнтів з поштовим сервером.

Налаштування аутентифікації через сервер OpenLDAP

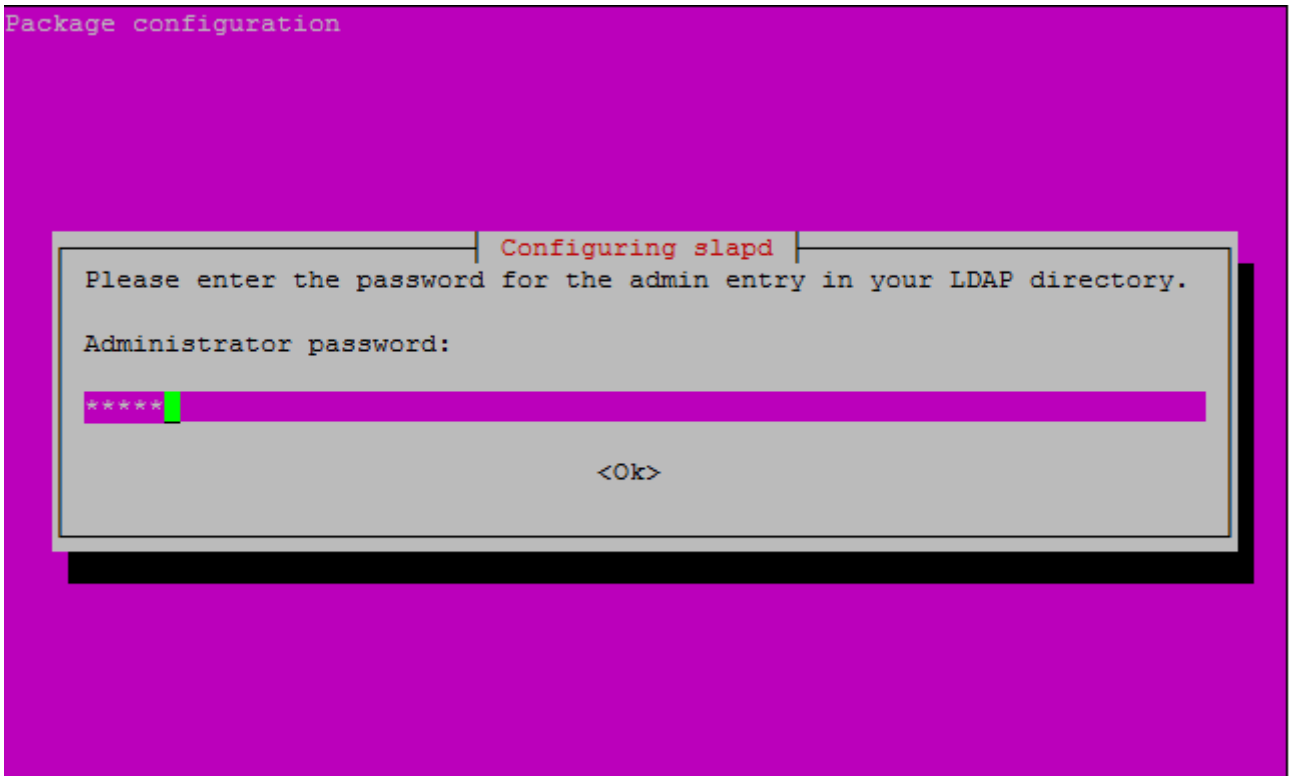
OpenLDAP - це відкрита реалізація LDAP, розроблена проектом OpenLDAP, поширюється під власною вільною ліцензією OpenLDAP Public License.
OpenLDAP складається з трьох головних компонентів:

- slapd - незалежний демон LDAP і відповідні оверлеї і інструменти;
- бібліотеки, що реалізують протокол LDAP;
- утиліти, інструменти та допоміжні клієнти

Встановлення сервера OpenLDAP

Для встановлення виконаємо команду

apt-get install slapd ldap-utils



Та на запит введемо адміністративний пароль.

Для роботи нам будуть потрібні два модуля. Перший для бази даних mdb, а другий модуль — monitor, який потрібен для створення та динамічної підтримки гілки про поточний статус демона slapd.

Для цього створимо файл add-mod.ldif та запишемо в нього

dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulePath: /usr/lib/ldap
olcModuleLoad: back_mdb.la
*olcModuleLoad: back_monitor.l*a

Далі виконаємо команду

ldapadd -QY EXTERNAL -H ldapi:/// -f add-mod.ldif

Додавання схем даних

Для подальшої роботи нам в OpenLDAP будуть потрібні наступні схеми:

- core.ldif
- cosine.ldif
- nis.ldif
- inetorgperson.ldif
- openldap.ldif
- misc.ldif

Якщо якоїсь схеми не вистачає, її можна підключити командою на зразок
ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/misc.ldif

Файли за необхідними схемами знаходяться в теці */etc/ldap/schema/*
Подивитись, які схеми вже підключені можна в теці */etc/ldap/slapd.d/cn=config/cn=schema*
Ініціалізація бази даних
Створимо власну базу даних для домену. Для цього робимо файл *db.ldif* та записуємо в нього

```
dn: olcDatabase=mdb,cn=config
objectClass: olcMdbConfig
olcDatabase: mdb
olcSuffix: dc=study,dc=local
olcDbDirectory: /var/lib/ldap
olcDbMaxsize: 1073741824
olcRootDN: cn=admin,dc=study,dc=local
olcRootPW: password
olcDbIndex: cn,sn,mail pres,eq,approx,sub
olcAccess: {0}to *
  by dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" manage
  by * break
olcAccess: {1}to attrs=userPassword
  by self write
  by anonymous auth
  by * none
olcAccess: {2}to *
  by self write
  by * read

dn: olcDatabase=monitor,cn=config
objectClass: olcDatabaseConfig
olcDatabase: monitor
```

Далі заносимо дані з нього до LDAP командою
ldapadd -QY EXTERNAL -H ldapi:/// -f db.ldif

Для зміни прав доступу створюємо файл *acl-mod.ldif* та заповнюємо його

dn: olcDatabase={-1}frontend,cn=config
changetype: modify
add: olcAccess
*olcAccess: {0}to **
by dn.exact=gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth manage
*by * break*
olcAccess: {1}to dn.base=""
*by * read*
olcAccess: {2}to dn.base="cn=subschema"
*by * read*

olcAccess: {1}to attrs=userPassword
by self write
by anonymous auth
*olcAccess: {2}to **
*by * read*

Далі командою
ldapadd -QY EXTERNAL -H ldapi:/// -f acl-mod.ldif
Вносимо ці дані до LDAP

Пересвідчимось, що обліковий запис адміністратора має доступ до служби каталогів:
ldapwhoami -WD cn=admin,dc=study,dc=local
Enter LDAP Password:

Тепер створимо дерево нашого домену створимо файл *tree.ldif* та занесемо в нього

dn: dc=study,dc=local
dc: study
objectClass: top
objectClass: domain

dn: ou=users,dc=study,dc=local
ou: Users
objectClass: top
objectClass: organizationalUnit
description: Central location for UNIX users

dn: ou=groups,dc=study,dc=local
ou: Groups
objectClass: top
objectClass: organizationalUnit
description: Central location for UNIX groups

І додамо ці дані командою
l*dapmodify -a -xWD cn=admin,dc=study,dc=local -f tree.ldif*

PhpLDAPAdmin

PhpLDAPadmin — це веб-додаток для адміністрування серверів Lightweight Directory Access Protocol (LDAP). Він написаний на мові програмування PHP, і розповсюджується під ліцензією GNU General Public License. Додаток доступний на 14 мовах і підтримує кодування UTF-8 для вмісту каталогу.

Встановимо веб-консоль керування LDAP командою:

apt-get install phpldapadmin

Для первинного налаштування в файлі конфігурації */etc/phpldapadmin/config.php* змінимо рядки

$servers→setValue('server','base',array('dc=example,dc=com'));

на

$servers→setValue('server','base',array('dc=study,dc=local'));

та

$servers→setValue('login','bind_id','cn=admin,dc=example,dc=com');

на

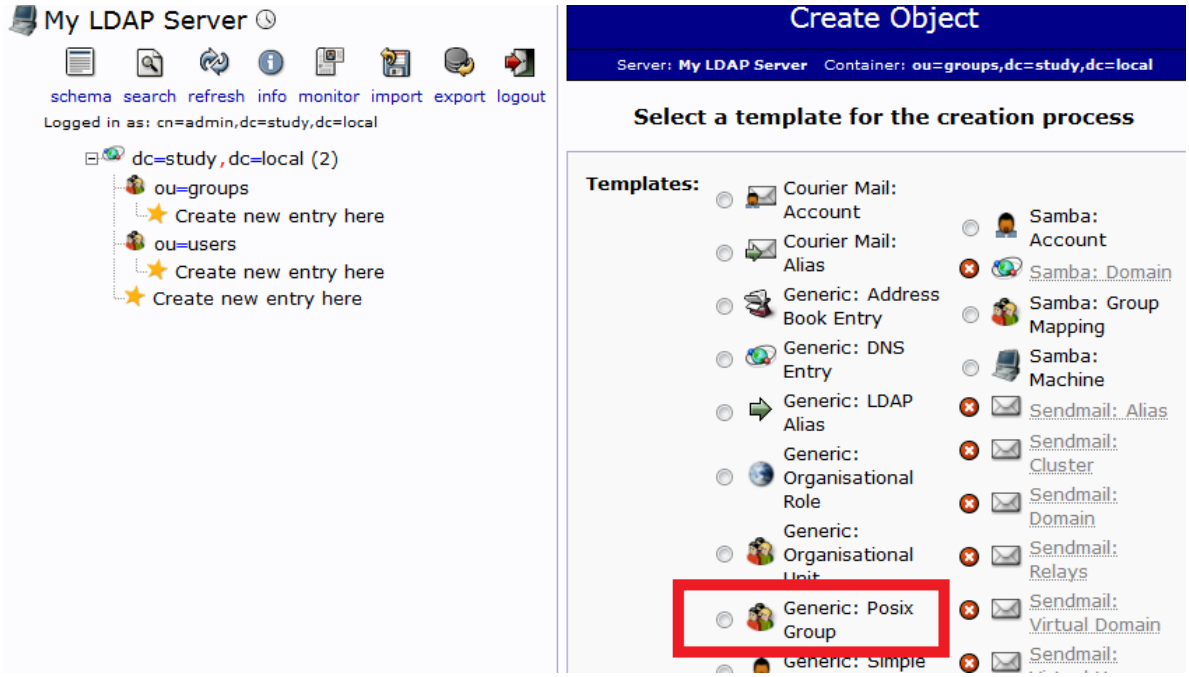
$servers→setValue('login','bind_id','cn=admin,dc=study,dc=local');

Створення користувачів та псевдонімів

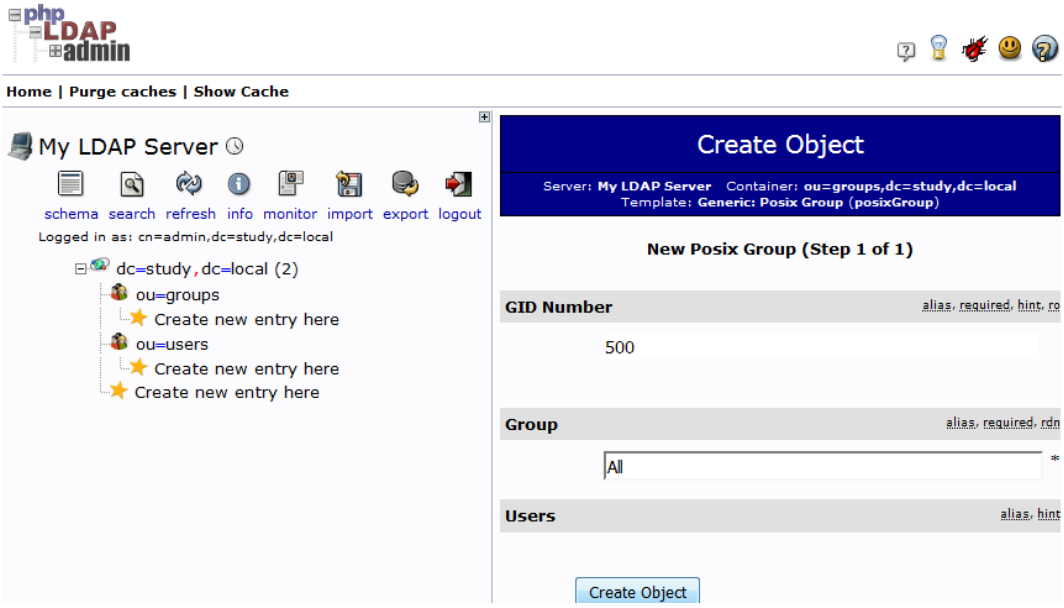
Підготовка

Перед тим, як створити будь якого користувача потрібно в контейнері groups створити групу з довільною назвою. Це пов'язано з тим, що будь який користувач має належати хоча б до однієї групи.

За посиланням http://server-ip/phpldapadmin заходимо в інтерфейс phpldapadmin. Далі переміщуємось у гілку users, та натиснувши на «Create new entry here» створюємо новий об'єкт типу “Generic: Posix Group”



Далі даємо назву нашій групі, наприклад All



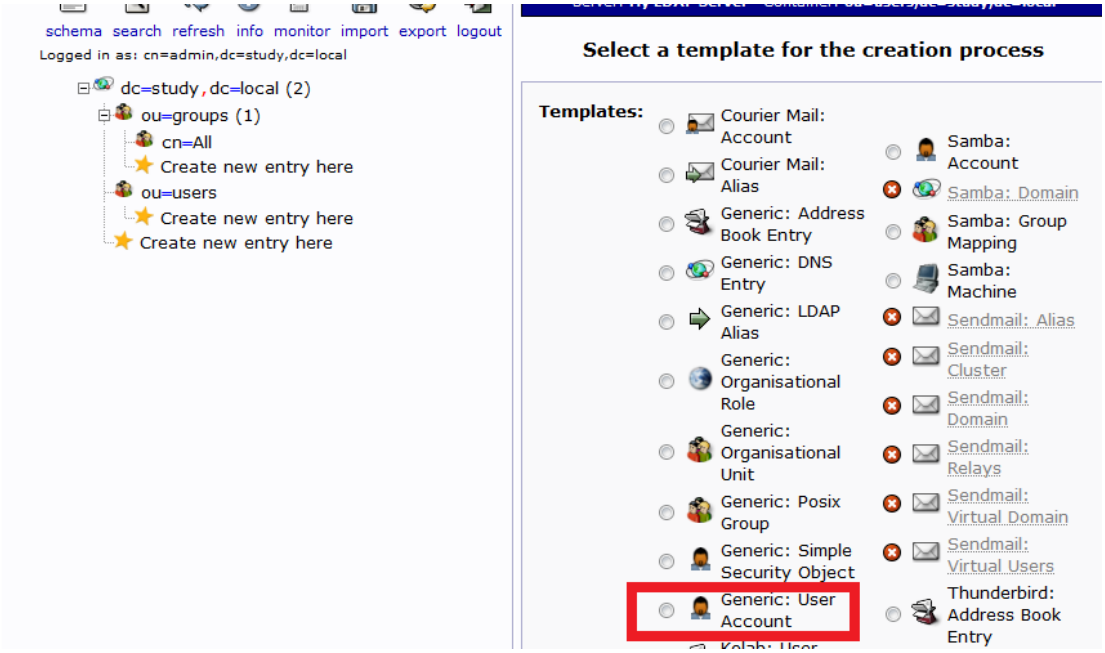
Наступним кроком підтверджуємо внесення даних до LDAP.

Створення користувачів

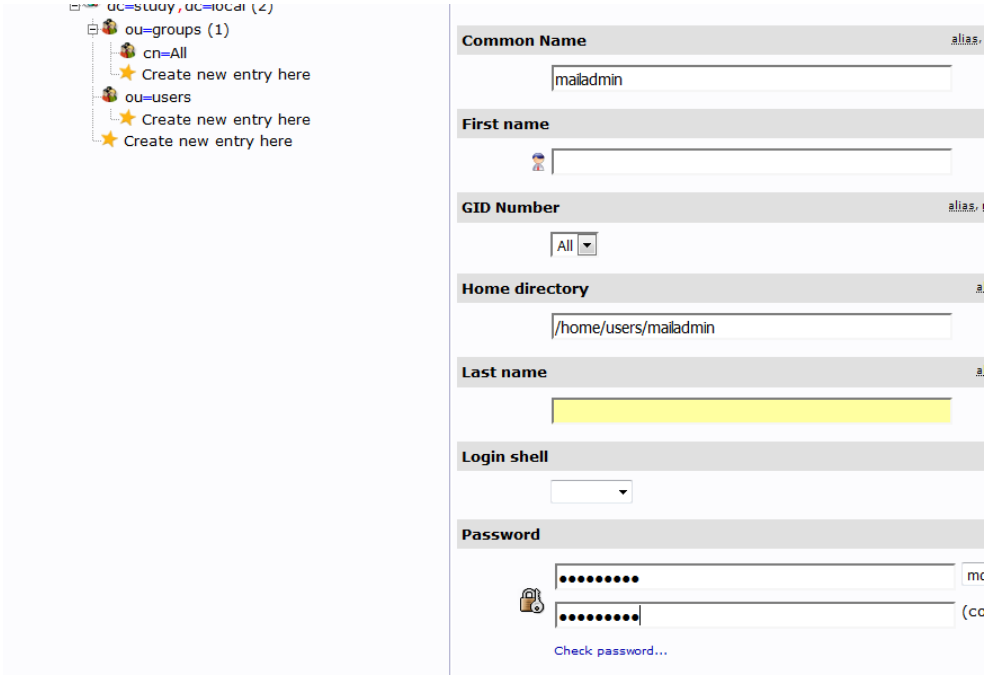
Тепер створимо нашого першого користувача — це буде користувач з іменем mailadmin, з правами якого наша поштова система буде вичитувати дані з LDAP

Для цього в розділі users тиснемо на «Create new entry here» створюємо новий об'єкт типу

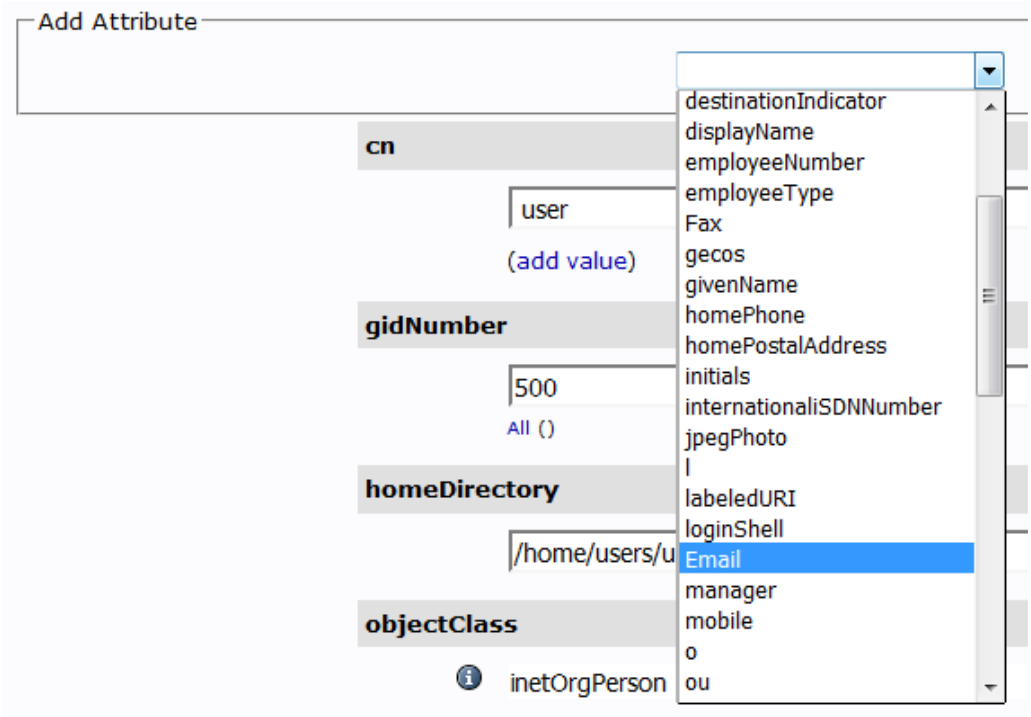
«Generic: User Account»



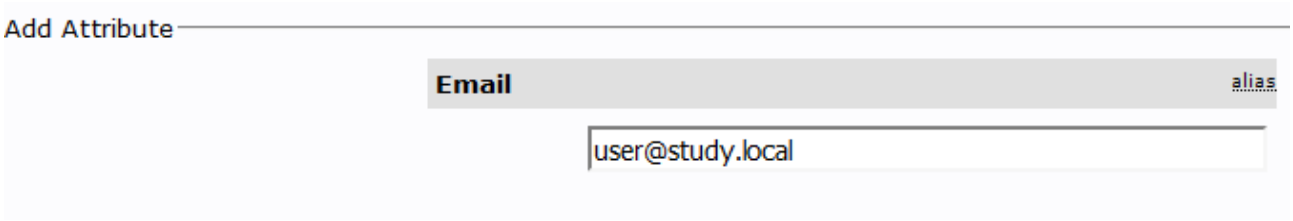
В наступному вікні заповнюємо всі обов'язкові поля та вводимо пароль. З усіх полів для нас важливими будуть Common Name та Password. Інші поля, навіть обов'язкові, можна заповнювати довільною інформацією.



Створимо першого користувача поштового сервера з іменем user. Він створюється так само, як і користувач mailadmin, але по закінченню потрібно вписати ще його email-адресу. Для цього згори вікна тиснемо на “Add new attribute” та у випадаючому списку додаємо атрибут email.



В новому полі вводимо поштову адресу користувача.



Аналогічним чином створюються і інші користувачі.

Створення псевдонімів.

Поштові псевдоніми зручніше за все створювати як групи користувачів. В інтерфейсі phpLDAPAdmin є шаблон об'єкту типу Generic: Posix Group, але нас цей шаблон не влаштовує. В ньому члени груп визначаються за uid, а Postfix останніх версій не вміє з uid отримати dn користувача. Тому ми будемо використовувати групи типу groupofnames.

Для цього можна створити файл gr.ldif і записати в нього:

dn: cn=mygroup,ou=groups,dc=study,dc=local
objectClass: groupofnames
objectClass: inetLocalMailRecipient
cn: mygroup
description: All users
member: cn=user,ou=users,dc=study,dc=local

Далі дані з цього файлу можна занести до LDAP і через веб-інтерфейс додати поле mailRoutingAddress і внести в нього поштову адресу псевдоніму командою.

ldapadd -x -D cn=admin,dc=study,dc=local -W -f gr.ldif

Але, на мою думку, набагато зручніше створити власний шаблон для phpLDAPAdmin і надалі користуватися саме ним. Для цього в теці */etc/phpldapadmin/templates/creation* створимо файл *groupOfNames.xml* з вмістом:

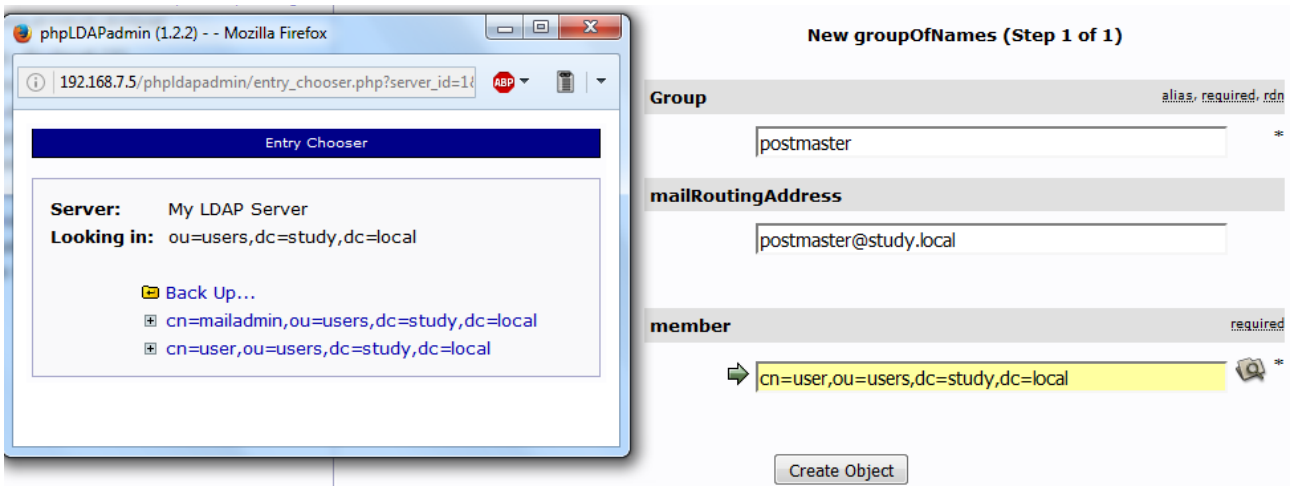
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE template SYSTEM "template.dtd">
<template>
<title>Mail Aliase Group</title>
<!-- <regexp>^ou=.*,</regexp> -->
<icon>images/ou.png</icon>
<description>New groupOfNames</description>
<askcontainer>1</askcontainer>
<rdn>cn</rdn>
<visible>1</visible>

<objectClasses>
<objectClass id="groupOfNames"></objectClass>
<objectClass id="inetLocalMailRecipient"></objectClass>
</objectClasses>
```

```
<attributes>
<attribute id="cn">
 <display>Group</display>
 <order>1</order>
 <page>1</page>
</attribute>
<attribute id="member">
 <display>member</display>
 <hint></hint>
 <order>2</order>
 <page>1</page>
 <spacer>1</spacer>
</attribute>
<attribute id="mailRoutingAddress">
 <display>mailRoutingAddress</display>
 <hint></hint>
 <order>3</order>
 <page>1</page>
 <spacer>1</spacer>
</attribute>
</attributes>
</template>
```

Після перелогіну в інтерфейсі з'явиться шаблон об'єкту з назвою “Mail Aliase Group”

При створенні групи за допомогою цього шаблону ми зразу можемо вказати email-адресу групи та внести до неї необхідних користувачів у потрібному нам форматі.



Налаштування Postfix

У файлі */etc/postfix/ldap_virtual_mailbox_maps.cf* записуємо:

```
server_host = 127.0.0.1
bind = yes
bind_dn = cn=mailadmin,ou=users,dc=study,dc=local
bind_pw = mailadmin
search_base = ou=users,dc=study,dc=local
```

query_filter = (&(mail=%s))
result_attribute = mail
result_format = %d/%u/

А в файл */etc/postfix/ldapalias*
server_host = 127.0.0.1
bind = no
bind_dn = cn=mailadmin,ou=users,dc=study,dc=local
bind_pw = mailadmin
search_base = dc=study,dc=local
query_filter = (&(objectclass=inetLocalMailRecipient)(mailRoutingAddress=%s))
special_result_attribute = member
leaf_result_attribute = mail

Перевірка коректності налаштувань Postfix

Після того, як ми зробили налаштування з'єднання з LDAP потрібно перевірити дані, які повернуться до Postfix.

Для перевірки роботи з поштовими скриньками виконаємо команду
postmap -q user@study.local ldap:/etc/postfix/ldap_virtual_mailbox_maps.cf
В результаті ми маємо отримати вивід
study.local/user/

Для того, щоб пересвідчитися в коректності роботи з псевдонімами виконаємо
postmap -q postmaster@study.local ldap:/etc/postfix/ldapalias
На виході має бути
user@study.local

Якщо дані повернулися правильні, то на цьому налаштування Postfix завершено.

Налаштування Dovecot

У файл */etc/dovecot/dovecot-ldap.conf.ext* записуємо
hosts = 127.0.0.1
auth_bind = yes
ldap_version = 3
base = dc=study,dc=local
dn = cn=mailadmin,ou=users,dc=study,dc=local
dnpass = mailadmin
deref = never
scope = subtree
user_attrs = uidNumber=5000,gidNumber=5000,mail=mail=maildir:/var/mail/%d/%n
user_filter = (&(objectClass=inetOrgPerson)(mail=%u))
pass_attrs = uidNumber=5000,gidNumber=5000,mail=mail=maildir:/var/mail/%d/%n
pass_filter = (&(objectClass=inetOrgPerson)(mail=%u))

Для перевірки налаштування Dovecot потрібно зв'язатися з сервером по протоколу IMAP за допомогою поштового клієнта, або утиліти telnet

Шифрування поштового трафіку

Не шифрувати трафік на поштовому сервері досить небезпечно. Пов'язано це не з перехопленням листів, а, перш за все, з тим, що зловмисник може перехопити логін та пароль одного з користувачів і використати цю інформацію для розсилання спаму.

Для шифрування використовуються SSL-сертифікати. Якщо в нас є можливість купити сертифікат від якогось центру сертифікації — то купуємо, якщо ж ні, то згенеруємо самопідписаний сертифікат. Це робиться командою:

openssl req -new -nodes -x509 -out smtpd.pem -keyout smtpd.pem -days 3650

Команда req змушує OpenSSL створити сертифікат.
Параметри цієї команди:
-new - створення запиту на сертифікат,
-nodes - не шифрувати закритий ключ,
-x509 (спільно з -new) - створити самоподпісаний сертифікат,
-keyout - задає місцезнаходження закритого ключа,
-out - задає місцезнаходження самоподпісаного сертифіката,
-days - задає час дії сертифіката (365x10 днів, що приблизно дорівнює десяти рокам).

В процесі виконання команди на екран будуть видані запити про введення таких параметрів як: Country Name, State or Province Name; Locality Name; Organization Name; Organizational Unit Name; Common Name; Email Address. Найважливішим параметром є значення Common Name. У нашому випадку воно повинно збігатися з FQDN-ім'ям сервера, по якому клієнти звертатимуться до сервера для відправлення та отримання пошти.

Шифрування трафіку в Postfix

Тепер налаштуємо роботу Postfix з сертифікатами
В файл /etc/postfix/main.cf додамо рядки

smtpd_tls_auth_only = yes
smtp_use_tls = yes
smtpd_use_tls = yes
smtpd_tls_cert_file=/etc/postfix/smtpd.pem
smtpd_tls_key_file=/etc/postfix/smtpd.pem
smtpd_tls_session_cache_database = btree:/var/lib/postfix/smtpd_scache
smtp_tls_session_cache_database = btree:/var/lib/postfix/smtp_scache
smtp_tls_note_starttls_offer = yes

Якщо при купівлі сертифікату вам видали кореневий довірений сертифікат, до у файл main.cf дописуємо рядок:

smtpd_tls_CAfile = /etc/postfix/root.crt

Пояснення параметрів

smtp_use_tls - використовувати TLS, якщо віддалений сервер повідомляє про підтримку TLS,

smtpd_use_tls - повідомляти клієнтам про підтримку TLS,

smtpd_tls_auth_only - використовувати аутентифікацію SMTP тільки для TLS-з'єднань, smtpd_tls_key_file - місцезнаходження закритого ключа сервера,

smtpd_tls_cert_file - місцезнаходження сертифіката сервера,
smtpd_tls_session_cache_database - файл в якому зберігається кеш tls-сесії
smtp_tls_note_starttls_offer - фіксувати в логах імена серверів, що видають повідомлення STARTTLS, підтримка TLS для яких не ввімкнена.
smtpd_tls_CAfile - місцезнаходження довіреного сертифікату

Вмикаємо SMTP submission
У файлі /etc/postfix/master.cf допишемо (або розкоментуємо) наступні рядки

```
submission inet n       -       -       -       -       smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
```

Адміністратори сервера обирають самі , який порт використовуватимуть клієнти для ретрансляції вихідної пошти - 25 або 587. Специфікації та багато серверів підтримують і той, і інший порти. Хоча деякі сервера підтримують порт 465 для безпечного SMTP, але краще використовувати стандартні порти та ESMTP-команди, у випадку коли необхідно встановити захищену сесія між клієнтом і сервером.

Відмінності портів 25, 465, 587. На 465 порті з'єднання відразу повинно відкриватися з шифруванням TLS/SSL. З портом 587 працюють так само як і з 25: з'єднання у відкритому вигляді, а для включення шифрування подається команда STARTTLS, якщо сервер заявив про таку можливість у відповідь на EHLO від клієнта. SMTPS (465 порт) більш старий стандарт, STARTTLS - новіший і, зрозуміло, більш гнучкий.

Налаштування роботи Dovecot.

Для dovecot можемо згенерувати свій сертифікат і ключ, а можемо використовувати той самий, що і для postfix. У випадку купівлі сертифікатів скоріш за все сертифікати будуть однаковими.

У файлі */etc/dovecot/conf.d/10-ssl.conf* прописуємо параметри

```
#Вмикаємо підтримку шифрування
ssl = yes
#Вказуємо шлях до файлів з закритим ключем та сертифікатом
ssl_cert = </etc/postfix/smtpd.pem
ssl_key = </etc/postfix/smtpd.pem
```

Після цих дій треба перезапустити postfix та dovecot
service postfix restart
service dovecot restart

Тепер наш поштовий сервер підтримує шифровані з'єднання і для клієнтів доступний порт 587 для відправлення електронних листів та порт 993 для шифрованого IMAP.

Встановлення поштового сервера у внутрішній мережі

Іноді поштовий сервер не має своєї білої IP-адреси. Він розгортається у внутрішній мережі, частіше за все у так званому DMZ-сегменті корпоративної мережі, а на інтернет-шлюзі, який має білу адресу, на поштовий сервер прокидуються відповідні порти. Всі налаштування, які були розглянуті раніше будуть цілком коректними, але з точки зору стандартів їх буде не достатньо. Це пов'язано з тим, що згідно стандартів сервер зобов'язаний приймати повідомлення на адреси типу user@[ip-address]. Postfix бачить лише свою власну сіру адресу і нічого не знає про білу, яка призначена інтернет-шлюзу. Для того щоб запобігти порушенню стандартів і примусити Postfix приймати листи і на такі адреси теж, в його файл конфігурації */etc/postfix/main.cf* слід додати рядок

proxy_interfaces = 1.2.3.4

де 1.2.3.4 — наша біла IP-адреса.

Робота з поштовим сервером за допомогою telnet

Після того, як ми налаштували поштовий сервер, треба перевірити його працездатність. Звичайно це можна зробити за допомогою поштового клієнта, але іноді простіше і швидше зробити це за допомогою утиліти telnet.

Поштовий сервер для відправлення листа використовує протокол SMTP, який за замовчуванням працює на tcp-порту 25. Цілком можливий, доречі, варіант його знаходження і на порту 587. Порт 587 використовує служба submission-SMTP з перевіркою достовірності, але це зовсім не означає що не буде проходити авторизація клієнтів поштового сервера на порті 25.

Якщо ми налаштували автентифікацію, то логін та пароль мають передаватися в кодованому вигляді, тому спочатку треба створити ці кодовані рядки.

$ perl -MMIME::Base64 -e 'print encode_base64("user\@study.local");'
bXlfbG9naW4K
$ perl -MMIME::Base64 -e 'print encode_base64("PASSWORD");'
bXlfcGFzc3dvcmQK

Саме ці значення ми будемо вводити замість логіна та пароля в нашій SMTP-сесії
В командному рядку відкриваємо нашу сесію
$ telnet mail-server 25
або, у випадку STARTTLS-сесії та порта 587
$ openssl s_client -starttls smtp -crlf -connect mail-server:587

Далі йде послідовність команд для відправлення листа з коментарями до них.

Trying 192.168.0.114... Connected to mail-srv. Escape character is '^]'.	На консолі відображується спроба з'єднання з сервером
220 mail.study.local ESMTP (ubuntu)	При вдалому з'єднанні сервер показує рядок вітання

ehlo test.com	Вводимо рядок вітання
250- mail-srv 250-PIPELINING 250-SIZE 10240000 250-VRFY 250-ETRN 250-AUTH PLAIN LOGIN 250-AUTH=PLAIN LOGIN 250-ENHANCEDSTATUSCODES 250-8BITMIME 250 DSN	Та отримуємо відповідь від сервера
auth login	Якщо потрібна авторизація на сервері - вводимо цей рядок. Якщо авторизація не потрібна, пропускаємо цю команду і продовжуємо введення з команди "mail from:"
334 VXNlcm5hbWU6	Отримуємо відповідь від сервера
bXlfbG9naW4K	Вводимо наш логін в форматі base64, який ми отримали на початку
334 UGFzc3dvcmQ6	Отримуємо відповідь від сервера
bXlfcGFzc3dvcmQK	Вводимо наш пароль в форматі base64, який ми отримали на початку
235 2.7.0 Authentication successful	Сервер повідомляє про успішну автентифікацію
mail from: user@test.com	Вказуємо адресу відправника листа
250 2.1.0 Ok	Відповідь від сервера про те, що адреса прийнята
rcpt to: yakim@study.local	Вказуємо адресу отримувача листа
250 2.1.5 Ok	Відповідь від сервера про те, що адреса прийнята
data	Після введення цієї команди починається сам лист
354 End data with <CR><LF>.<CR><LF>	Отримуємо відповідь від сервера
subject: test telnet auth	Якщо потрібна тема листа - ввести цю команду
test .	Тут пишемо сам лист. Він ОБОВ'ЯЗКОВО повинен закінчуватися послідовністю ENTER . ENTER
250 2.0.0 Ok: queued as 415211810C6	Відповідь від сервера про те, що лист прийнято та поставлено до черги
quit	Вводимо команду закінчення роботи
221 2.0.0 Bye Connection closed by foreign host.	Сервер сповіщає про закінчення сесії

Після цих дій, лист буде прийнято сервером і відправлено на адресу отримувача.

Тепер перевіримо роботу IMAP-сервера.
Скористаємося для цього тією ж утилітою telnet.
З'єднаємося з сервером
$ telnet mail-server 143

Якщо у нас з'єднання з сервером IMAP зашифроване (по SSL), то команда з'єднання буде наступною:
$ openssl s_client -crlf -ign_eof -connect mail-server:993

Далі йде послідовність команд для роботи з сервером з коментарями до них.

Trying 192.168.0.114... Connected to mail-srv. Escape character is '^]'.	На консолі відображується спроба з'єднання з сервером
* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE AUTH=PLAIN AUTH=LOGIN] Dovecot ready	При вдалому з'єднанні сервер показує рядок вітання
. login our-login our-password	Після команди . login вводимо в тому ж рядку логін і пароль у відкритому вигляді
. OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE SORT SORT=DISPLAY THREAD=REFERENCES THREAD=REFS MULTIAPPEND UNSELECT CHILDREN NAMESPACE UIDPLUS LIST-EXTENDED I18NLEVEL=1 CONDSTORE QRESYNC ESEARCH ESORT SEARCHRES WITHIN CONTEXT=SEARCH LIST-STATUS] Logged in	Сервер повідомляє про успішний логін
. list "" "*"	Вводимо команду перегляду списку поштових тек
* LIST (\HasNoChildren) "." "Drafts" * LIST (\HasNoChildren) "." "Spam" * LIST (\HasNoChildren) "." "Trash" * LIST (\HasNoChildren) "." "Sent" * LIST (\HasNoChildren) "." "INBOX" . OK List completed.	Сервер виводить список тек
. status INBOX (messages)	Запитуємо у сервера статус теки Inbox
* STATUS "INBOX" (MESSAGES 1086) . OK Status completed.	Сервер виводить статус теки
. select inbox	Обираємо для подальшої роботи теку inbox
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft $MDNSent Junk NonJunk $Forwarded KMAILFORWARDED KMAILTODO KMAILWATCHED	Відповідь сервера

KMAILIGNORED $TODO $WATCHED $IGNORED receipt-handled $label2 $has_cal) * OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft $MDNSent Junk NonJunk $Forwarded KMAILFORWARDED KMAILTODO KMAILWATCHED KMAILIGNORED $TODO $WATCHED $IGNORED receipt-handled $label2 $has_cal *)] Flags permitted. * 1086 EXISTS * 0 RECENT * OK [UNSEEN 1085] First unseen. * OK [UIDVALIDITY 1242120321] UIDs valid * OK [UIDNEXT 138212] Predicted next UID * OK [HIGHESTMODSEQ 187388] Highest . OK [READ-WRITE] Select completed.	
. fetch 7 full	Даємо команду серверу показати лист № 7
* 7 FETCH (FLAGS (\Seen) INTERNALDATE "12-Jul-2008 18:24:12 +0300" RFC822.SIZE 1935 ENVELOPE ("Sat, 12 Jul 2008 18:07:52 +0300 (EEST)" "test mail" (("test.com" NIL "isbear" "ukrpost.net")) (("test.com" NIL "isbear" "ukrpost.net")) (("test.com" NIL "isbear" "ukrpost.net")) ((NIL NIL "yakim" "test.com.net")) NIL NIL NIL "< 20080712150752.42B10207527AB@web.test.org.ua>") BODY ("text" "plain" ("charset" "KOI8-U") NIL NIL "7bit" 279 8)) . OK Fetch completed.	Сервер показує заголовки листа
. fetch 7 rfc822.text	Даємо команду серверу показати тіло листа № 7
Re: test mail this is test mail	Сервер показує тіло листа
. logout	Вводимо команду закінчення роботи
* BYE Logging out . OK Logout completed. Connection closed by foreign host	Сервер сповіщає про закінчення сесії

Налаштування поштового антивірусу ClamAV

Встановимо антивірусну систему для поштового сервера:
#apt-get install clamsmtp

Відкриємо файл конфігурації */etc/clamsmtpd.conf* та запишемо туди потрібні параметри:
OutAddress: 10026
Listen: 127.0.0.1:10025

ClamAddress: /var/run/clamav/clamd.ctl
Header: X-AV-Checked: ClamAV using ClamSMTP
TempDirectory: /var/spool/clamsmtp
PidFile: /var/run/clamsmtp/clamsmtpd.pid
Quarantine: on
User: clamsmtp
#VirusAction: /etc/clamav/script.sh

Насправді, параметрів в цьому файлі може бути більше, але тут вказані лише необхідні. Для більш детального вивчення рекомендую почитати man clamsmtpd.conf.

Для застосування змін необхідно перезапустити сервіс антивірусу:
#service clamsmtp restart

Налаштування поштового сервера для роботи з антивірусом

В файл */etc/postfix/main.cf* додамо 2 рядки:
content_filter = scan:[127.0.0.1]:10025
receive_override_options = no_address_mappings

Перша повідомляє postfix'у про те, що необхідно пересилати всю пошту через сервіс (фільтр) scan на 10025-ий порт, на якому слухає clamsmtpd. Другий рядок каже, щоб postfix не робив ніяких маніпуляцій з адресами до того, як вони дійдуть до content_filter. Так що виходить, що фільтр працює з реальними поштовими адресами, а не з результатами переведення у віртуальні псевдоніми, маскарадингом і т.п.

В файл /etc/postfix/master.cf необхідно додати такі рядки:
 # AV scan filter (used by content_filter)
scan unix - - n - 16 smtp
-o smtp_send_xforward_command=yes
 # For injecting mail back into postfix from the filter
127.0.0.1:10026 inet n - n - 16 smtpd
-o content_filter=
-o receive_override_options=no_unknown_recipient_checks,no_header_body_checks
-o smtpd_helo_restrictions=
-o smtpd_client_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipient_restrictions=permit_mynetworks,reject
-o mynetworks_style=host
-o smtpd_authorized_xforward_hosts=127.0.0.0/8

Лишилось перезапустити сервіс Postfix:
#service postfix restart

На цьому базове налаштування антивірусу завершене.

Налаштування повідомлень антивірусу

Тепер залишилася остання дія — налаштувати відсилання повідомлень антивірусу. Для цього створимо файл script.sh:

#nano /etc/clamav/script.sh

Та запишемо в нього:

```
#!/bin/sh
DOMAIN=study.local
# Email address to send alerts to
ADMIN=postmaster@study.local
# formail should be in PATH
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin
LINE="-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-"
if [ X`echo $SENDER | egrep $DOMAIN` != "X" ];
then MAILTO=$SENDER,$ADMIN
else MAILTO=`echo "$RECIPIENTS" | egrep $DOMAIN | tr '\n' ','`$ADMIN
fi
(echo "Virus name: $VIRUS"
echo "Sender: $SENDER"
echo "Recipient(s): $RECIPIENTS"
echo
if [ "x$EMAIL" != "x" ] && [ -f $EMAIL ]
then
echo "Quarantined to: $EMAIL"
fi
) | cat -v | mail -s "$VIRUS found on mailserver" $MAILTO
```

Розкоментуємо рядок *VirusAction: /etc/clamav/script.sh* у файлі */etc/clamsmtpd.conf* та перезапустимо сервіс clamsmtp:

#service clamsmtp restart

Зараз у нас антивірус не тільки перевіряє пошту, але й складає заражені листи в карантин, відсилаючи повідомлення про це адміністратору і користувачам нашого домену.

Для відправлення повідомлень треба встановити пакет mailutils

 # apt-get install mailutils

Антиспам SpamAssassin

Встановлення SpamAssassin

Для встановлення антиспаму виконаємо команду

 # apt-get install spamassassin

Зразу після встановлення програма працювати не буде. За замовчанням вона відключена. Для її активації треба в файлі */etc/default/spamassassin* змінити значенння *enable* в 1.

Підключення SpamAssassin до Postfix

Вносимо зміни в файл /etc/postfix/master.cf.
Зразу після рядка
smtp inet n - - - - smtpd
Додаємо
-o content_filter=spamassassin

Не забуваємо, що цей новий рядок має відступати від початку рядка
Цим параметром ми вказуємо, що всі листи мають передаватися на аналіз до фільтра spamassassin

Тепер в кінці цього файлу треба, власне, описати цей фільтр. Для цього допишемо рядки

spamassassin unix - n n - - pipe
user=virtual argv=/usr/bin/spamc -f -e /usr/sbin/sendmail -oi -f ${sender} ${recipient}

Файл конфігурації SpamAssassin

Основний файл конфігурації антиспама — це /etc/spamassassin/local.cf. Приводимо його до вигляду:

*rewrite_header Subject *****SPAM*****
report_safe 0
trusted_networks 192.168.0.0/24
required_score 5.0
use_bayes 1
bayes_auto_learn 1
bayes_ignore_header X-Bogosity
bayes_ignore_header X-Spam-Flag
bayes_ignore_header X-Spam-Status
bayes_min_ham_num 1
bayes_min_spam_num 1
report_charset koi8-r
ok_locales ru en uk

bayes_path /var/spool/bayes/bayes
bayes_file_mode 0666

score SUBJ_FULL_OF_8BITS 0
score FROM_ILLEGAL_CHARS 0
score SUBJ_ILLEGAL_CHARS 0
score HEAD_ILLEGAL_CHARS 0
score HABEAS_SWE 0
score FORGED_IMS_TAGS 1
score BAYES_00 0.0001 0.0001 -2.312 -2.599
score BAYES_05 0.0001 0.0001 -1.110 -1.110
score BAYES_20 0.0001 0.0001 -0.740 -0.740
score BAYES_40 0.0001 0.0001 -0.185 -0.185
score BAYES_50 0.0001 0.0001 0.001 0.001
score BAYES_60 0.0001 0.0001 2.0 2.0*

score BAYES_80 0.0001 0.0001 3.0 3.0
score BAYES_95 0.0001 0.0001 3.5 3.5
score BAYES_99 0.0001 0.0001 5.0 5.0
score ALL_TRUSTED -3.360 -3.440 -3.665 -3.800

Основні параметри
rewrite_header показує який рядок додає в тему листа
trusted_networks 192.168.0.0/24 довірена мережа звідки перевірка спаму не проводиться
required_score 5.0 поріг спрацьовування на спам, за замовчуванням 5 балів, якщо 5 балів та більше, то лист позначається як спам
use_bayes 1 вмикаємо алгоритм bayes (самонавчання)
ok_locales ru en uk список припустимих мов

Параметри оцінювання листів на належність до спаму, а також кількість балів за кожне спарювання вказується після ключового слова score. Докладніше про ці параметри можна дізнатися з документації на сайті розробників за посиланням
http://spamassassin.apache.org/tests_3_3_x.html

Правил може бути будь-яка кількість і вони додаються в стовпчик. Для кожного правила призначається певна кількість балів. Правила, які не описані в цьому файлі, працюватимуть з кількістю балів за замовчуванням. Тобто всі правила, які знаходяться за посиланням працюють, а в файлі ми прописуємо тільки ті правила, для яких ми змінюємо кількість балів. Для відключення правила йому потрібно призначити 0 балів.

Створення білих списків адрес

Білий список адрес заповнюється адресами, з яких точно не може прийти спам. В файлі */etc/spamassassin/local.cf* прописуємо параметр *whitelist_from* і далі робимо список адрес. Наприклад:
*whitelist_from user@mail.ru *@gmai.com*

Всі адреси записуються через пробіл. При записі можна використовуватися регулярні вирази. Для зручності,щоб не робити один довгий рядок, таких рядків можна зробити багато.

Навчання антиспаму

Механізм самонавчання - це алгоритм bayes. Це не частина статичних правил spamassassin, а окремий алгоритм. Додаткові бали він додає або знімає виходячи з власного досвіду навчання на базі листів і робить це динамічно. При створенні бази листів для навчання потрібно бути уважним, тому що базу навчання легко зіпсувати помилково доданими листами. З точки зору розробників ні в якому разі не можна включати до бази навчання на спам (--spam) листи які вже позначені як спам, а в навчанні на помилкове спрацьовування (--ham) не можна включати листи не позначені як спам.

Будь-яка система антиспаму може як пропускати спамові листи, так і генерувати помилкові спрацьовування (тобто нормальний лист може позначитись як спам). Якщо потрібно передати в навчальний механізм неопізнані спамові листи (система не позначила їх як спам, хоча це точно спам) то використовується команда:
/usr/bin/sa-learn --spam
для навчання на листах з помилковим спрацьовуванням використовується команда:
/usr/bin/sa-learn --ham

Автоматизація навчання антиспаму

Вручну навчати антиспам не продуктивно, тому зробимо автоматизацію процесу.
У кожному поштовому ящику створюємо дві теки

1. Spam - в цю теку користувач вручну складає листи не виявлені антиспамом
2. Nospam - в цю теку користувач вручну складає листи помилково позначені як спам

У теці /root створюємо дві підтеки — spam і nospam

Слід звернути увагу що в файловій системі теки Spam і Nospam в поштовій скринці користувача (/var/mail/domain/user) будуть називатися відповідно .Spam і .Nospam, це пов'язано з нюансами роботи dovecot
Крім цього потрібно враховувати, що листи зберігаються не в самій теці (наприклад .Spam), а в підтеках .Spam/cur (прочитані листи) та .Spam/new (непрочитані листи)

Далі пишемо скрипт для автоматизації процесу.

В теці /root створюємо файл скрипта spam.sh і записуємо в нього:

```
#!/bin/bash
MAILDIR=/var/mail/study.local     #визначимо в змінній теку зберігання пошти
#Spam
for filename in $MAILDIR/* # Обхід всіх файлів в теці.
do
if [ -d $filename ]; then
   if [ -e $filename/.Spam ]; then
      ssp=$filename'/.Spam/cur/';
      ssp2=$filename'/.Spam/new/';
      sp1=`ls $ssp`

     if [ "$sp1" != "" ];
      then
      mv -f $ssp/* /root/spam && chmod 777 -R /root/spam;
      mv -f $ssp2/* /root/spam && chmod 777 -R /root/spam;
    fi;
  fi;
fi;
done
date >>/var/log/spam-learn.log
/usr/bin/sa-learn --spam /root/spam >>/var/log/spam-learn.log

MDIR=/var/mail/peopleandlaw.ru
#NoSpam
for filename in $MDIR/* # Обхід всіх файлів в теці.
do
if [ -d $filename ]; then
if [ -e $filename/.Nospam ]; then
ssp=$filename'/.Nospam/cur/';
ssp2=$filename'/.Nospam/new/';

sp1=`ls $ssp`
```

```
if [ "$sp1" != "" ];
then
mv -f $ssp/* /root/nospam && chmod 777 -R /home/root/nospam;
mv -f $ssp2/* /root/nospam && chmod 777 -R /home/root/nospam;
fi;
fi;
fi;
done
date >>/var/log/spam-learn.log
/usr/bin/sa-learn --ham /root/nospam >>/var/log/spam-learn.log
```

Тепер встановлюємо на цей файл право на виконання
chmod +x /root/spam.sh

Та дописуємо в планувальник cron (файл */etc/crontab*) правило періодичного запуску цього скрипта
*10 1 * * * root /root/spam.sh*

Тепер цей скрипт буде виконуватися щодоби вночі о 1 годині 10 хвилин

Налаштування сірих списків (GreyListing)

Робота сірих списків базується на тому, що спамери при помилці відправлення частіше за все другий раз лист не відправляють, а легальні поштові сервера будуть намагатися відправляти лист протягом не менше ніж двох діб.

При першому прийомі листа наш сервер повертає помилку 450 і розриває сесію, тобто лист не приймаємо через тимчасову помилку.

Після закінчення заданого часу очікування (за замовчуванням 300 секунд) лист буде прийнято і адреса сервера відправника буде внесена до тимчасового білого списку. За замовчуванням в цьому списку сервер знаходиться 35 діб з моменту останньої вдалої сесії.

Для початку встановимо необхідне програмне забезпечення
 # apt-get install postgrey

Налаштування Postfix

Правимо конфиг postfix
для цього в файлі *main.cf* в блок *smtpd_recipient_restrictions* додамо перевірку, тобто допишемо наступний рядок:
check_policy_service inet:127.0.0.1:10023

краще його ставити відразу після рядка *check_sender_access*

Налаштування Postgrey

Основний файл конфігурації це */etc/default/postgrey*
Приведемо параметр POSTGREY_OPTS до вигляду

POSTGREY_OPTS = "--inet=127.0.0.1:10023 --delay=20 --max-age=60 --whitelist-clients=/etc/postgrey/whitelist_clients"

де

--inet=127.0.0.1:10023 postgrey слухає з'єднання на відповідному ip адресу і порті

--delay=20 затримка в секундах перед прийомом листа, тобто перший лист буде відкинуто, а другий, якщо піде через 20 і більше секунд, буде прийнято

--max-age=60 кількість днів перебування в білому списку

--whitelist-clients=/etc/postgrey/whitelist_clients файл з білим списком серверів відправника

Налаштування постійного білого списку серверів відправників

Білий список серверів міститься у файлі /etc/postgrey/whitelist_clients В цьому файлі кожен сервер відправника записується в окремому рядку. Це може бути або доменне ім'я, або регулярний вираз, які відповідають доменному імені, або IP-адреса. Приклад частини файлу:

1.2.3.4
mail.testdomain.com
/.\.domain\.com$/*

Налаштування DNS для роботи поштового сервера

Для коректної роботи поштового сервера треба правильно налаштувати DNS.

В описі нашої доменної зони треба, як мінімум, зробити MX-записи. Також бажано прописати зворотню зону, SPF, DKIM та DMARC

Основні налаштування DNS

Для того, щоб інші сервера знали, що саме наш сервер приймає пошту для нашого домену, в описі зони треба зробити наступні налаштування:

1. Зробити A-запис для нашого сервера (пряма зона)
2. Зробити MX-запис з вказанням приорітету та нашим доменним іменем
3. Надіслати заявку провайдеру, хостеру, чи іншій організації яка надала нам IP-адресу, з проханням прописати зворотню зону (PTR-запис) для нашої IP-адреси з нашим іменем з прямої зони

Налаштування SPF

SPF запис — це запис в DNS, в якому вказується, які саме сервера мають право відправляти пошту від імені даного домену.

SPF — це TXT запис, який знаходиться в налаштуваннях DNS зони. Наприклад:

"v = spf1 +a + mx +a:mail.study.local ~all"

Пояснення вмісту запису:

- + дозволено

- - заборонено
- ~ можна, але не бажано
- v = spf1 використовуємо запис версії 1
- +a можна відправляти пошту з A запису (тобто з основної адреси домену)
- +mx можна відправляти пошту з серверів, для яких прописані MX (тобто поштові сервери даного домену)
- +a:mail.study.local можна відправляти пошту з хоста mail.study.local (можна вказувати хост не з цього домену)
- ~all від інших пошту приймати можна, але довіри система виставить менше.

Налаштування DKIM

DKIM - це цифровий підпис, який поштовий сервер вставляє в кожен лист на підставі закритого ключа openpgp. Відкритий ключ зберігається в DNS і сервер одержувача перевіряє валідність відправника на підставі відкритого ключа і цифрового підпису.

Налаштування DKIM на сервері Postfix

Встановимо необхідні програми

apt-get install opendkim opendkim-tools

Далі створюємо теку /etc/mail і переходимо в неї. Там ми генеруємо ключі командою *opendkim-genkey -t -s mail -d study.local*

де

mail — це так званий селектор ключа (ім'я) береться довільно
study.local — домен для якого створюємо пару ключів

В результаті команди в теці */etc/mail* з'являться два файли *mail.txt* і *mail.private* — це і є ключі.

Далі редагуємо файл */etc/default/opendkim*
Вписуємо в нього рядок
SOCKET="inet:8891@localhost"
тут ми вказуємо, де буде слухати з'єднання opendkim.

Файл */etc/opendkim.conf* приводимо до вигляду

```
Syslog                  yes
UMask                   002
Domain          study.local          # ім'я домену який буде верифікуватися
KeyFile         /etc/mail/mail.private
Selector        mail     #селектор заданий при генерації ключа
AutoRestart         yes
Background           yes
Canonicalization      relaxed/relaxed
```

DNSTimeout 5
Mode sv
SignatureAlgorithm rsa-sha256
SubDomains no
X-Header no
OversignHeaders From
Statistics /var/log/dkim-filter/dkim-stats #файл логу dkim

Додаємо рядки в кінець файла */etc/postfix/main.cf*
 # DKIM
milter_default_action = accept
milter_protocol = 2
smtpd_milters = inet:localhost:8891 # рядок має співпадати з рядком /etc/default/opendkim
non_smtpd_milters = inet:localhost:8891 # рядок має співпадати з рядком /etc/default/opendkim

Налаштування DKIM в DNS

В DNS для нашого домена додаємо новий запис типу TXT та іменем — mail._domainkey. Значення цього запису — це вміст файлу */etc/mail/mail.txt*, тобто відкритий ключ для нашого домену.

Для файлу опису зони bind9, наприклад, цей рядок буде виглядати так:

mail._domainkey IN TXT ("v=DKIM1; k=rsa; t=y; "
"p=MIGfMA5GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDhL5g8W+AVPOgiiZyPdayinqMwCa
sbDh06K9Ixy5D575iYWQZYMGyzlDlUIJLqLDBBCFxRRs36tb4p/EY0OkmyHNisK/Y4cB/joVnQm
K/7XVwkQt1GVNuzodRjcQPXGFx5VEaQi7+O54gEF2eOEyGt/FPWG882AFAFpkrJekuiawIDAQ
AB")

Тепер треба перезапустити postfix і через деякий час, коли оновляться DNS-записи, листи від нашого сервера можна буде перевіряти через DKIM.

Налаштування DMARC

Після створення записів SPF і DKIM необхідно налаштувати перевірку DMARC, додавши в записи DNS домену правила у вигляді TXT-запису.

DMARC задає політику, як перевіряти пошту в домені і що робити, якщо лист не проходить перевірку SPF або DKIM.

Базовий запис DMARC виглядає так:
_dmarc.study.local IN TXT "v = DMARC1; p =;"

p - policy - політика, може бути:

- none - не приймати ніяких особливих дій, все на розсуд поштового сервера;
- quarantine - відправити в спам;
- reject - не приймати лист.

Але таке налаштування підходить тільки в разі одиничного сервера. Більш правильна політика, яка враховує наявність піддоменів з яких може слатися пошта і дозволяє отримувати звіти виглядає так:

_dmarc. study.local IN TXT "v=DMARC1; p=none; sp=none; rua=mailto:postmaster@study.local"

sp - subdomain policy - може приймати значення ті ж що і policy;

rua - reporting URI for aggregate reports - задає поштову адресу в форматі mailto:mbox@ study.local на який раз на добу будуть приходити звіти в XML

Керування поштовими повідомленнями в Dovecot

Для керування листами в Dovecot включена підтримка Sieve.

Sieve — це мова опису правил фільтрації для поштових повідомлень. Створена компанією Cyrusoft International, Inc./ISAMET під час роботи над поштовим сервером Cyrus. За допомогою написання скриптів з правилами обробки пошта є можливість:

1. автоматично сортувати листи по поштовим текам виходячи з певних ознак (адреса відправника, тема та ін.)
2. автоматично видаляти листи виходячи з певних ознак
3. налаштувати автоматичну відповідь (vacation) на певні листи

Обробка листів має відбуватися перед тим, як лист потрапляє в поштову скриньку користувача. В тих налаштуваннях агентом локальної доставки (LDA – local delivery agent) виступає сам Postfix. Для підключення розширення Sieve цей функціонал треба перекласти на Dovecot.

Налаштування Dovecot як LDA

Для роботи Dovecot в якості LDA треба встановити пакет підтримки lmtp командою apt-get install dovecot-lmtpd

В самому dovecot налаштовувати нічого не треба.

Далі налаштовуємо Postfix. Вносимо зміни в конфіг postfix

В файл /etc/postfix/main.cf додаємо рядки:

#як локальний транспорт (для протоколу lmtp) будемо використовувати фільтр під #назвою dovecot (ім'я фільтра вибираємо довільно)
virtual_transport = dovecot
dovecot_destination_recipient_limit = 1 #доставляємо листи по одному

Також в файлі /etc/postfix/master.cf потрібно описати фільтр під назвою dovecot (ім'я задано раніше в файлі master.cf). Для цього в кінець файлу додаємо рядки опису нашого фільтру

 # Dovecot LDA
dovecot unix - n n - - pipe
flags = DRhu user = virtual: virtual argv = /usr/lib/dovecot/deliver -d $ {recipient}
#доставка листів буде проводиться з правами virtual:virtual за допомогою утиліти

#/usr/lib/dovecot/deliver

Після цих налаштувань у нас з'явилася можливість підключити sieve.

Підключення розширення Sieve до Dovecot

Встановлюємо для dovecot розширення sieve
apt-get install dovecot-sieve dovecot-managesieved
dovecot-sieve — цей пакет дає можливість використовувати функціонал sieve
dovecot-managesieved — цей пакет дає можливість клієнту створювати правила фільтрації самостійно.

Файл конфігурації dovecot-managesieved залишаємо без змін, так як він нас влаштовує і приступаємо до налаштування самого sieve.

Спочатку в основному файлі конфігурації dovecot включимо підтримку sieve. Для цього в файлі */etc/dovecot/dovecot.conf* рядок protocols приводимо до вигляду
protocols = imap sieve

В налаштуваннях lda підключаємо підтримку sieve, для цього в файлі */etc/dovecot/conf.d/5-lda.conf* блок protocol lda приводимо до вигляду:

protocol lda {
 # Space separated list of plugins to load (default is global mail_plugins).
mail_plugins = $ mail_plugins sieve
}

Підключаємо цей же плагін до протоколу lmtp. Для цього в файлі */etc/dovecot/conf.d/20-lmtp.conf*
блок protocol lmtp приводимо до вигляду
protocol lmtp {
 # Space separated list of plugins to load (default is global mail_plugins).
mail_plugins = $ mail_plugins sieve
info_log_path = /var/log/dovecot-lmtp.log
}

Тепер налаштовуємо місце розташування скриптів правил sieve для користувачів. Для цього в файлі */etc/dovecot/conf.d/90-sieve.conf* змінюємо відповідні рядки і приводим їх до вигляду:

sieve = /etc/dovecot/sieve/%Ld/%n/sieve/%u.sieve # Розташування активного скрипта
sieve_dir = /etc/dovecot/sieve/%Ld/%n/sieve
 # Каталог для скриптів — це бібліотека скриптів з правилами

Для коректної роботи обидва параметри повинні бути унікальні для кожного користувача. При вказуванні шляху можна використовувати внутрішні змінні Dovecot:

- %Ld — це частина логіна користувача, що відповідає поштовому домену
- %n — це частина логіна користувачадо собаки, тобто імя клієнта
- %u — це повний логін користувача

Після всіх цих дій, обробка і створення правил може бути виконана лише адміністратором поштового сервера шляхом правки скриптів. Набагато зручніше дати користувачам можливість самим створювати власні правила.

Підключення плагіна managesieve в roundcube

Спочатку встановимо набір плагінів для RoundCube командою

 # apt-get install roundcube-plugins

Далі йдемо в */etc/roundcube/plugins/managesieve* і бачимо, що файл конфігурації порожній, але в ньому є вказівка, де взяти його зразок.

Виконуємо команду

cp /usr/share/roundcube/plugins/managesieve/config.inc.php.dist /etc/roundcube/plugins/managesieve/config.inc.php

Цією командою створюємо файл конфігурації зі зразка і починаємо налаштування.

Знаходимо відповідні рядки та змінюємо їх відповідно до нашого сервера:

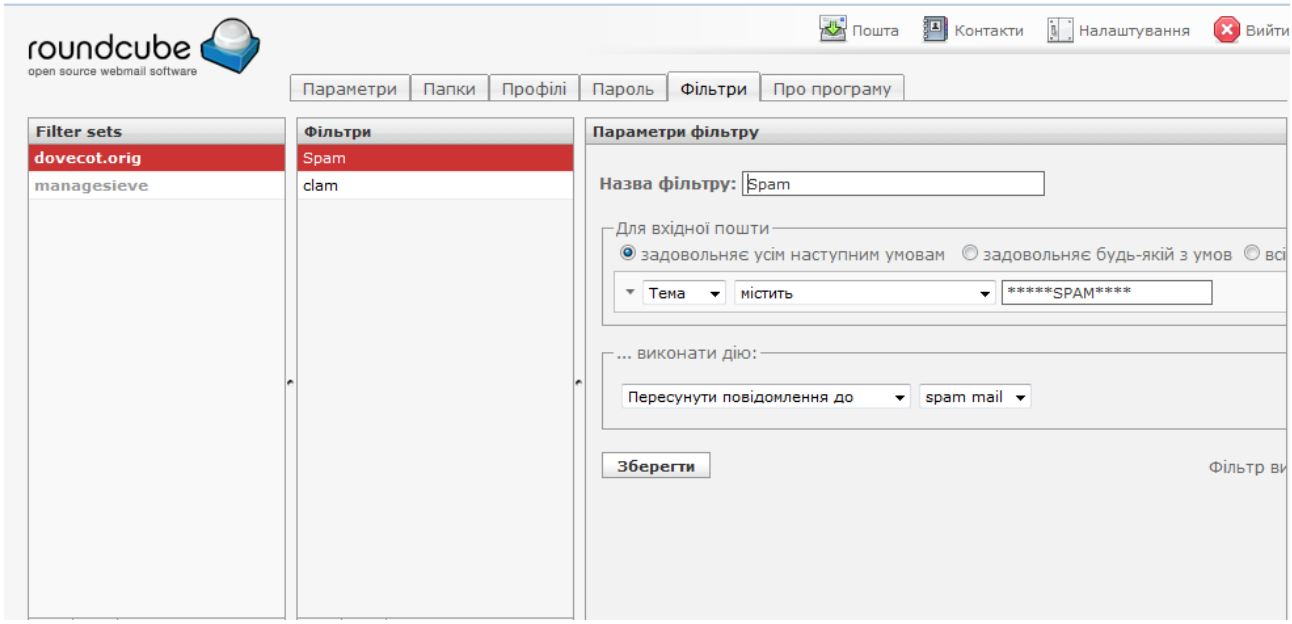
$rcmail_config['managesieve_port'] = 4190; #вказуємо порт для сервера managesieve
$rcmail_config['managesieve_host'] = 'localhost'; # вказуємо де знаходиться сервер #managesieve
$rcmail_config['managesieve_auth_type'] = login; # вказуємо тип авторизації на сервері. #Цей параметр беремо з налаштувань dovecot
$rcmail_config['managesieve_usetls'] = false; #відключаємо підтримку шифрування

Далі в файлі /etc/roundcube/main.inc.php підключаємо плагін managesieve.

Для цього знаходимо рядок і приводимо до вигляду:

$Rcmail_config ['plugins'] = array ('managesieve');

Після цих налаштувань в RoundCube в інтерфейсі користувача на сторінці налаштувань з'являється закладка filters, де і створюються правила обробки пошти



Підтримка плюс-адресації на поштовому сервері

Якщо є необхідність підтримки плюс-адресації на поштовому сервері, то необхідно буде доналаштувати нашу систему.

В файл конфігурації Postfix - */etc/postfix/main.cf* слід додати рядок

recipient_delimiter = +

В тому випадку, якщо у нас Dovecot виступає в ролі LDA, треба зробити додаткові налаштування. По перше в файлі /etc/dovecot/conf.d/15-lda.conf слід додати або розкоментувати рядок

recipient_delimiter = +

А в файлі */etc/postfix/master.cf* потрібно змінити наш фільтр dovecot.

Він має виглядати наступним чином

 # Dovecot LDA

dovecot unix - n n - - pipe

flags=DRhu user=virtual:virtual argv=/usr/lib/dovecot/dovecot-lda -f ${sender} -a $original_recipient} -d ${user}@${nexthop}

Поштові адреси — які вони бувають

Звичайні адреси

Зазвичай адреси електронної пошти виглядають наступним чином:
user@domain.com

Така адреса складається з двох частин — локальної частини, яка стоїть перед символом «@» та доменної частини, яка знаходиться після цього символу. Доменна частина повинна відповідати Fully Qualified Domain Name (FQDN).

Адреси без доменної частини, на зразок:
user
не можуть використовуватись. Листи, що приходять на такі адреси повинні відкидатися поштовим сервером.

Єдиний вийняток — адреса *postmaster.* Будь поштовий сервер повинен приймати пошту на цю адресу і доставити її кому-небудь, хто відповідає за цю поштову систему. (Див RFC2821 глава 4.5.1).

Локальна частина адреси повинна інтерпретуватися лише хостом, зазначеним в доменній частини адреси. Цей формат адреси визначено в RFC2821 та RFC2822.

Плюс-адресація

Знак плюса ("+") — це один з допустимих символів в адресах електронної пошти згідно RFC-5233.

Адреса, в такому випадку, буде виглядати так:

user+detail@domain.com

де detail — абсолютно довільна частина. Таким чином користувач сам створює собі потрібну кількість поштових псевдонімів, готових до використання в будь-який момент. При доставці повідомлення в поштову скриньку ліва частина адреси коротшає і від неї відкидається все, починаючи з символу «+», тобто лист з будь-яким detail потрапить в поштову скриньку користувача.

Коментарі в адресах

Так само стандарти допускають використання коментарів прямо в адресі. Наприклад:

user(comment)@domain.com

(comment)user@domain.com

user@(comment)domain.com

user@domain.com(comment)

Щоправда на практиці таку адресацію підтримує дуже невелика кількість поштових серверів.

Address literals

Якщо існує (тимчасова) проблема з системою DNS, для адресації може використовуватися літерація адрес (вона також називається доменною літерацією). В такому випадку поштова адреса буде виглядати наступним чином:

user@[10.11.12.13]

Використання доменної літерації було сильно обмежено в RFC822, але в новому RFC2821 нічого про неї не говориться. Такі адреси можна час від часу бачити вживу. Слід зазначити, що для такого роду адрес обов'язково використовувати квадратні дужки.

Маршрути Source Route

Спеціальна форма запису email адреси може визначати source route. Природно, такий запис — це більше ніж просто адреса. Це адреса з доданням інформації про маршрутизацію листа, в якій зазначено, через які сервера повинен пройти лист на шляху від відправника до одержувача. Адреса source route виглядає так:

@dom1.com,@dom2.edu:user@domain.com

Це означає, що лист буде відправлений на поштовий сервер домену dom1.com, далі він відправиться на dom2.edu і лише потім буде пересланий на адресу user@domain.com Сьогодні цей формат є застарілим і більшістю поштових серверів не підтримується. Пов'язано це з масовим розсиланням спаму.

(Див RFC2821, RFC822, RFC1123)

Хак з відсотком

Так само для source route існує так званий хак з відсотком. У такому випадку адреса буде виглядати наступним чином:

user%domain.com%dom2.edu@dom1.com

Так само як і в попередньому випадку лист буде відправлено на поштовий сервер домену dom1.com, далі він відправиться на dom2.edu і лише потім буде пересланий на адресу user@domain.com. Транзитний поштовий сервер при пересиланні повинен відкидати частину адреси починаючи з "@" і замінити останній символ «%» на «@».

Даний спосіб так само вважається застарілим і не використовується на практиці у зв'язку з небезпекою спам-розсилань.

Зверніть увагу, що немає ніякого офіційного документа, який робить знак відсотка особливим. Цей функціонал залежить виключно від обробки адрес сервером, що приймає лист.

Адресація у форматі UUCP

Колись давно люди обмінювалися повідомленнями за допомогою UUCP (Unix To Unix Copy).

Повідомлення часто передавалися через декілька серверів. Тоді ще не було централізованої системи DNS і ніхто не міг дізнатися адреси всіх серверів у мережі. Якщо ви хотіли передати комусь повідомлення, то ви повинні були знати всі хости між вашим комп'ютером і одержувачем. Поштова адреса мала вигляд:

serv1!serv2!serv3!user

Тут прописано зліва направо через які сервери пройде лист, доки він дійде до одержувача user. Остання частина адреси тут є не іменем хоста, а адресою користувача.

Сьогодні ще теоретично можна зустріти таке написання адреси, але зараз така адресація вже вважається неприпустимою.

Так само існує проблема змішування UUCP і сучасних адрес. наприклад

serv1!serv2!serv3!user@domain.com

Не існує офіційного правила, як такі адреси мають оброблятися.

Адресація X.400

X.400 є стандартом пошти, розробленим ISO. На сьогодні він відіграє лише другорядну роль, але вже є компанії, що використовують його у своїй внутрішній поштовій системі з виходом в Інтернет.

Поштова адреса у форматі X.400 буде виглядати наступним чином:

S=postmaster; OU=it-department; P=office; A=domain; C=ua;

Як видно, вона не має ніякої схожості зі звичайними адресами. Такі адреси використовують ієрархію атрибутів і значень. Тут 'S' означає 'surname', 'OU' - 'organisational unit', 'C' - 'country' і таке інше. Атрибути не завжди однакові.

Для взаємодії з іншими поштовими серверами адреси трансформуються в звичайний вигляд. Наприклад наступним чином:

S=postmaster/OU=it-department/P=office/A=domain/C=ua@domain.ua

Список використаної літератури

1. http://www.wikipedia.org/ — різними мовами
2. https://habrahabr.ru/ — велика кількість статей з різних нюансів налаштування серверів
3. http://opennet.ru — там взагалі багато цінної інформації
4. http://wiki2.dovecot.org/
5. http://www.postfix.org
6. https://roundcube.net/
7. http://yakim.org.ua — ну як же я міг не взяти матеріали з власного сайту.