# Elastyczny system poczty elektronicznej dla serwerów wirtualnych

Marcin Sochacki
*wanted@linux.gda.pl*

**Pingwinaria**, Szczytno, 2001

**Streszczenie**

Referat dotyczy problemu, z którym boryka się większość ISP — poszukują oni elastycznego i uniwersalnego sposobu zarządzania dużą ilością kont pocztowych dla wielu klientów.

**Vmail-SQL**, czyli rozwiązanie jakie zastosowałem bazuje na programach: *Exim* (demon SMTP), *tpop3d* (demon POP3) oraz bazie danych MySQL, jako centralnym miejscu przechowywania listy kont. System obsługuje wirtualne domeny, zarówno bazujące na różnych IP, jak też różniące się tylko nazwą w DNS (działające na wspólnym numerze IP).

## 1 Wstęp

Niniejszy artykuł jest sprawozdaniem z mojej pracy jako administratora serwerów Uniksowych. Docelowym odbiorcą są inni Uniksowcy, a zatem od Czytelnika wymagam pewnego poziomu wiedzy. Wiele zagadnień (celowo) potraktowałem powierzchownie w przekonaniu, że średnio zaawansowany administrator bez trudu poradzi sobie z detalami. Pomysł na podzielenie się wiedzą i doświadczeniem z innymi Linuksowcami powstał dość spontanicznie. Akurat „na tapecie" miałem zadanie stworzenia systemu pocztowego obsługującego wielu niezależnych klientów. Dodatkowym problemem była konieczność płynnego i niezauważalnego od strony użytkownika przemigrowania kont z poprzedniego serwera, z tradycyjnym Uniksowym systemem pocztowym.

Dlaczego konieczne jest użycie specjalnego oprogramowania do opisanego celu? Standardowe narzędzia uniksowe nie spełniają podstawowych wymogów przedsięwzięcia:

- obsługa wirtualnych domen (na wspólnym i na różnych numerach IP),
- obsługa kont o tych samych nazwach w różnych domenach, np. jan@klient1.pl, jan@drugi.klient.pl,
- dowolność w wyborze nazwy użytkownika: długość, znaki interpunkcyjne,
- łatwe zarządzanie kontami, m.in. przez WWW,
- wolnodostępny kod.

Przejrzałem co nieco istniejące rozwiązania i wybór padł na *Vmail-SQL* autorstwa Chris'a Lightfoot'a z Wielkiej Brytanii. Chris jest osobą, w moim mniemaniu, w pełni zasługującą na miano *hackera*. Mam na myśli oczywiście *hackowanie* w pozytywnym sensie zdefiniowanie przez Eric'a S. Raymond'a w *The Jargon File*[1]. Strona WWW[2] Chris'a uzmysławia jak wieloma projektami się zajmował, zarówno poważnymi, jak i pisanymi wyłącznie dla przyjemności.

Dlaczego akurat *Vmail-SQL* zdobył przychylność? Pierwszym warunkiem była współpraca z moim preferowanym MTA[3] — *Exim'em*. Ten pakiet jest wręcz dedykowany dla *Exim'a*, choć dopasowanie go do innego nowoczesnego MTA powinno być w miarę łatwe.

Popularność systemu *Vmail-SQL* jest faktycznie niewielka. Mam nadzieję, że moje doświadczenia w przecieraniu szlaków zaowocują większą liczbą instalacji na polskich serwerach. Jest takie angielskie przysłowie — „*Variety is the spice of life*", co w wolnym tłumaczeniu oznacza „*Urozmaicenie to prawdziwy smak życia*". Dlatego gorąco zachęcam do testowania mało znanych, ale ciekawych i obiecujących projektów.

# 2 Przygotowania

Każdy system obsługujący e-mail musi być wyposażony w niezbędne elementy: baza użytkowników oraz usługa wysyłania i odbierania poczty. Bazę użytkowników w typowym systemie stanowi plik `/etc/passwd`. To tradycyjne podejście nie sprawdza się jednak przy dużych ilościach kont dla wielu klientów.

[1] http://www.tuxedo.org/~esr/jargon/html/entry/hacker.html
[2] http://www.ex-parrot.com/~chris/
[3] ang. *Mail Transport Agent*.

## 2.1 Idea działania systemu

Transport e-maili w Internecie jest obecnie podporządkowany protokołowi SMTP[4]. Sposobów czytania poczty przez użytkownika jest co najmniej kilka: z poziomu *shell'a* Uniksowego, przez protokół POP3[5] czy też IMAP[6].

Centralną częścią systemu *Vmail-SQL* jest baza danych w MySQL, w której przechowywane są informacje o kontach. Takie podejście wymusza odpowiednią interakcję między usługami sieciowymi a bazą danych. W momencie otrzymania poczty serwer SMTP musi zweryfikować w bazie, na jakim koncie umieścić przesyłkę. Podobnie, serwer POP3 podczas połączenia przeprowadza autoryzację użytkownika na podstawie loginu i hasła zapisanego w MySQL.

Hasła przechowywane są w postaci *hash'a* MD5, choć od niedawna jest możliwość wprowadzenia do bazy haseł wygenerowanych tradycyjną funkcją `crypt` w Uniksie. Dzięki temu można przeprowadzić łatwą migrację użytkowników z systemu bazującego na */etc/passwd*.

Obsługa serwerów wirtualnych zawsze nastręcza dużo problemów w przypadku poczty. O ile odpowiednie mechanizmy są obecne w protokole HTTP czy SMTP, to niestety POP3 nie przewidywał obsługi kilku domen na jednym adresie IP. Aby ominąć to ograniczenie stosuje się dwie metody:

- **przydzielanie osobnych IP dla wszystkich domen** — najczęściej z wykorzystaniem aliasingu. Powoduje to „zmarnowanie" jednego adresu IP na każdą obsługiwaną domenę, co przy obecnych oszczędnościach przestrzeni IPv4 jest niebagatelną stratą.

- **jedno IP dla wszystkich domen, rozbudowany login** — polega na podawaniu **całego** adresu e-mail w polu na nazwę użytkownika programu pocztowego. Dzięki domenie występującej po „@" serwer może dokonać właściwej autoryzacji użytkownika.

Większość czytników poczty obsługuje bezproblemowo nazwy użytkowników w formie `luser@klient1.pl`. Dość popularny Netscape Messenger ma zwyczaj obcinania nazwy po znaku „@", w takim programie wprowadzamy login w następującej postaci: `login!klient1.pl`.

[4]ang. *Simple Mail Transport Protocol*
[5]ang. *Post Office Protocol*, wersja 3.
[6]ang. *Internet Message Access Protocol*.

Poza tym drobnym technicznym ograniczeniem pozostanie inne — psychologiczne. Musimy przekonać klientów do nowego schematu autoryzacji, co nie zawsze jest łatwe. Jeśli migrujemy z tradycyjnego systemu i celem nadrzędnym jest zachowanie ustawień w programach klientów, powinniśmy poświęcić osobne IP na obsługę domeny.

Przejdźmy do sedna sprawy, czyli instalacji odpowiedniego oprogramowania.

## 2.2 Serwer SMTP: *Exim*

*Exim* jest moim ulubionym demonem SMTP; jego podstawowe zalety to:

- licencja GPL,
- pełna i drobiazgowa dokumentacja,
- czytelny dla przeciętnego śmiertelnika plik konfiguracyjny `exim.conf`,
- zgodność z RFC i innymi standardami Internetu,
- bogactwo funkcji, m.in. wbudowana obsługa LDAP, MySQL, PostgreSQL,
- łatwość kompilacji na różnorodnych systemach, również przestarzałych,
- niezła szybkość działania.

W większości dystrybucji Linuksa znajdziemy *Exima* bez problemu w postaci gotowych, binarnych pakietów (np. *.deb*, *.rpm*). Jednak do naszych celów nie zawsze owe gotowce będą w 100% dopasowane. Przede wszystkim potrzebujemy wsparcia dla MySQL, które np. w dystybucji Debian Potato nie jest domyślnie wkompilowane.

Ze strony http://www.exim.org/ lub jej najbliższego mirrora możemy ściągnąć pakiet źródłowy. Po rozpakowaniu wystarczy wyedytować plik `src/EDITME`, a następnie skopiować go jako `Local/Makefile`. Oprócz typowego dostrojenia Makefile'a do swojego systemu, należy pamiętać o odkomentowaniu linii:

```
LOOKUP_MYSQL=yes
```

Potem wystarczy standardowa sekwencja `make; make install` aby otrzymać gotowego do działania Eximka :-)

## 2.3 Serwer POP3: *tpop3d*

Autorem *tpop3d* jest sam Chris Lightfoot, który napisał ten program specjalnie na potrzeby *Vmail-SQL*. Wcześniej wykorzystywał on istniejący pakiet *gnu-pop3d*[7] dodając swojego patch'a do obsługi MySQL. Z biegiem czasu jednak poziom jego irytacji wzrósł na tyle, że stwierdził: „napiszę własny serwer POP3". Jak powiedział, tak uczynił.

Do podstawowych zalet i funkcji *tpop3d* należy zaliczyć:

- szybkość działania — z łatwością obsługuje wielomegabajtowe skrzynki pocztowe (oczywiście potrzebne są też odpowiednie zasoby sprzętowe),
- zwarty, przemyślany i bezpieczny kod,
- łatwość rozbudowy — istnieje proste API umożliwiające tworzenie własnych schematów autoryzacji użytkownika,
- elastyczna i przejrzysta konfiguracja,
- obsługa różnych formatów skrzynek pocztowych (*mailbox*, *maildir*),
- kilka schematów autoryzacji — `/etc/passwd`, PAM, APOP, MySQL,
- obsługa domen wirtualnych,
- możliwość włączenia rozbudowanych logów na etapie testowania.

Pracę z programem rozpoczynamy od ściągnięcia źródła z: http://www.ex-parrot.com/~chris/tpop3d/. W chwili pisania tego artykułu najnowszą dostępną wersją jest `tpop3d-1.3.2.tar.gz`.

Po rozpakowaniu musimy podjąć decyzję co do opcji, które chcemy wkompilować. Czynimy to w standardowy sposób — podając parametry do skryptu `configure`. Należy pamiętać o tym, że większość opcji nie koliduje ze sobą; właściwego wyboru dokonujemy w pliku konfiguracyjnym już po kompilacji programu. Nie widzę zatem przeszkód (może poza oszczędzaniem zasobów) aby wkompilować od razu jak najwięcej usług.

- Autoryzacja

  Dostępne są następujące mechanizmy autoryzacji:

[7] http://www.nodomainname.net/software/mailutils/

**auth_pam** (`--enable-auth-pam`)

Użycie PAM-a, czyli *Pluggable Authentication Modules*. Na większości nowszych Uniksów PAM jest podstawowym narzędziem do autoryzacji.

**auth_passwd** (`--enable-auth-passwd`, `--enable-shadow-passwords`)

Prosta autoryzacja poprzez standardowe Uniksowe pliki `passwd` i `shadow`.

**auth_mysql** (`--enable-auth-mysql`, `--with-mysql-root`)

Lista użytkowników pobierana z bazy danych w MySQL — ta opcja jest najbardziej interesująca dla ISP. Opcja `--with-mysql-root` wskazuje na katalog główny naszej instalacji MySQL'a. Do połączenia z bazą wykorzystywany jest klient `libmysqlclient.so`. Ponadto kompilator musi mieć dostęp do pliku nagłówkowego `mysql.h`. W standardowej dystybucji Debiana, jeśli instalujemy MySQL z pakietów, należy wprowadzić:

```
--with-mysql-root=/usr
```

**auth_other** (`--enable-auth-other`)

Autoryzacja jest powierzana zewnętrznemu programowi; można w ten sposób rozwijać własne mechanizmy przechowywania listy użytkowników.

**auth_perl** (`--enable-auth-perl`)

Wkompilowanie wewnętrznego interpretera Perl'a i autoryzacja poprzez procedury napisane w tym języku. Opcja ta może istotnie zwiększyć wielkość pliku wyjściowego *tpop3d*.

- Formaty skrzynek pocztowych

  **BSD mailbox** (`--enable-mbox-bsd`)

  Standardowy format Uniksowy, cała poczta w jednym pliku.

  **Maildir** (`--enable-mbox-maildir`)

  Format wprowadzony w *Qmailu* i coraz częściej używany w innych systemach pocztowych. Każdy list przechowywany jest w oddzielnym pliku.

- Blokowanie (*locking*) skrzynek

Blokowanie ma na celu uniknięcie sytuacji, gdy dwa lub więcej procesów próbowałoby zapisywać do jednego pliku. Taka sytuacja mogłaby się zdarzyć, gdy np. użytkownik czyta pocztę przez POP3 i jednocześnie otrzymuje nowy list. Blokowanie stosuje się tylko w przypadku formatu *BSD mailbox*. Niestety wraz z rozwojem Uniksa pojawiło się wiele niekompatybilnych mechanizmów blokowania. *tpop3d* obsługuje większość (wszystkie?) z nich.

**fcntl** (`--enable-fcntl-locking`)

Mechanizm blokowania obsługiwany przez wszystkie nowsze systemy. Działa również na plikach podmontowanych sieciowo przez NFS[8].

**flock** (`--enable-flock-locking`)

Starszy mechanizm z BSD, nie działa przez NFS.

**dotfile** (`--enable-dotfile-locking`)

Prymitywne blokowanie pliku poprzez dodanie koncówki „.lock" do jego nazwy. W miarę przyzwoicie działa poprzez NFS.

**C-Client** (`--enable-cclient-locking`)

*C-Client* to biblioteka napisana na Uniwersytecie w Waszyngtonie, wykorzystywana m.in. w programie *pine*. *tpop3d* może „podkradać" blokadę skrzynki pocztowej od *pine'a*.

Dla większości systemów skrypt `configure` powinien automagicznie wykryć dostępne sposoby blokowania.

- Dodatkowe opcje

**„Wredne komentarze"** (`--disable-snide-comments`)

Opcja ta służy do wyłączenia dość uszczypliwych komunikatów wyświetlanych przez serwer POP3 w reakcji na błędy użytkownika. Dla przykładu:

```
$ telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK <a27de2fd90af52ed630ff096b2fcd7c1@debian>
```

[8] *Network File System*.

```
user wanted
+OK Tell me your password.
pass blabla
-ERR Lies! Try again!
$
```

Niektórzy mogliby poczuć się dotknięci takimi komunikatami, choć autor twierdzi, że „użytkownicy powinni przyzwyczaić się do tego, że komputery są dla nich niegrzeczne" :-)

Włączenie tej opcji przywróci standardowe, „grzeczne" komunikaty.

W mojej konfiguracji wybieram następujące opcje:

```
./configure --enable-auth-pam --enable-auth-passwd \
            --enable-shadow-passwords --enable-auth-mysql
```

Następnie standardowo odpalamy `make; make install`, co powinno umieścić plik wykonywalny `tpop3d` w katalogu `/usr/local/sbin`. Warto go jeszcze przed użyciem „zestripować", czyli usunąć z pliku binarnego zbędne symbole służące głównie diagnostyce: `strip /usr/local/sbin/tpop3d`. Moj plik wykonywalny zajmuje zaledwie 70 KB.

# 3 Konfiguracja

Zakładając, że mamy już gotowe środowisko do uruchomienia systemu, zajmijmy się konfiguracją.

Na początek ściągamy pakiet *Vmail-SQL* z: http://www.ex-parrot.com/~chris/vmail-sql/. W chwili pisania tego dokumentu najnowszą wersją jest `vmail-sql-0.3.tar.gz`. Pakiet zawiera opis konfiguracji systemu oraz skrypty do obsługi bazy z poziomu shella i WWW.

## 3.1 Dodajemy konto Uniksowe

Do obsługi wszystkich „wirtualnych" kont wystarczy jeden Uniksowy użytkownik, np. *vmail*. Konto oczywiście powinno być zablokowane, z powłoką `/bin/false`. Odpowiednie linie w `/etc/passwd` i `/etc/shadow` powinny wyglądać podobnie do poniższych:

```
vmail:x:106:65534::/home/vmail:/bin/false
vmail:!:11392:0:99999:7:::
```

Wykrzyknik w polu z hasłem uniemożliwia zalogowanie się na dane konto w tradycyjny sposób, tzn. telnet, ssh itp.

## 3.2 Zakładamy bazę danych w MySQL

Używając standardowych narzędzi MySQL'a zakładamy bazę, np. o nazwie *virtualemail* oraz użytkownika, któremu nadajemy wybrane hasło i prawa do operacji `SELECT`, `INSERT`, `UPDATE` i `DELETE`:

```
$ mysql -p -u root
Enter password:
mysql> CREATE DATABASE virtualemail;
Query OK, 1 row affected (0.00 sec)
mysql> USE mysql;
mysql> INSERT INTO user (Host,User,Password)
       VALUES('localhost','vmail',password('sekret'));
mysql> INSERT INTO db (Host,Db,User,Select_priv,Insert_priv,
       Update_priv,Delete_priv)
       VALUES('localhost','virtualemail','vmail','Y','Y','Y','Y');
mysql> FLUSH PRIVILEGES;
```

Oczywiście sposób zakładania bazy i użytkownika jest zupełnie dowolny — można np. skorzystać z SQL-owych instrukcji `GRANT`.

Kolejnym krokiem jest założenie tabel według wskazówek w pliku `INSTALL`. Główną tabelą jest *domain*, zawierająca listę domen wirtualnych obsługiwanych przez nasz system. Każdej domenie przyporządkowano hasło i maksymalną liczbę kont, ponieważ jest możliwe zarządzanie kontami przez osobę z zewnątrz. Tabela *popbox* zawiera listę kont we wszystkich domenach, a *forwarder* umożliwia przekierowywanie poczty, na kształt sendmail'owego *virtuserstable*. Strukturę bazy przedstawiono w tabeli 1.

## 3.3 Edytujemy `exim.conf`

Jak już wspomniano *Exim* obsługuje MySQL praktycznie od ręki. Wystarczy tylko dopisać kilka linijek do pliku konfiguracyjnego. W `exim.conf` wyróżnione są następujące sekcje:

1. *Main configuration settings*

| **Nazwa pola** | **Typ** | **Opis** |
|---|---|---|
| Tabela *domain* | | |
| domain_name | varchar(255) | nazwa domeny |
| unix_user | varchar(255) | użytkownik Uniksowy przypisany do domeny |
| password_hash | varchar(255) | zaszyfrowane hasło do zarządzania domeną |
| path | varchar(255) | ścieżka do katalogu, w którym przechowywane są skrzynki pocztowe |
| max_popbox | int(11) | maksymalna liczba kont możliwych do założenia w danej domenie |
| Tabela *popbox* | | |
| domain_name | varchar(255) | nazwa domeny |
| local_part | varchar(255) | użytkownik (część przed „@") |
| password_hash | varchar(255) | zaszyfrowane hasło użytkownika (zwykle MD5) |
| mbox_name | varchar(255) | nazwa pliku ze skrzynką pocztową (zwykle taka sama jak *local_part*) |
| Tabela *forwarder* | | |
| domain_name | varchar(255) | nazwa domeny |
| local_part | varchar(255) | użytkownik (część przed „@") |
| remote_name | varchar(255) | docelowy adres |
| Tabela *web_session* | | |
| domain_name | varchar(255) | nazwa domeny |
| session_id | varchar(32) | identyfikator sesji |
| expires | int(11) | czas wygaśnięcia sesji |

Tabela 1: Struktura bazy *virtualemail*.

2. *Transports configuration*
3. *Directors configuration*
4. *Routers configuration*
5. *Retry configuration*
6. *Rewrite configuration*

Sekcje oddzielone są słowami kluczowymi `end`. W większości przypadków nie ma znaczenia kolejność wpisywania komend w ramach sekcji, z wyjątkiem *directors* i *routers*.

### 3.3.1 `exim.conf`: Sekcja *main*

Musimy wskazać *Exim'owi* bazę danych, z którą ma nawiązać połączenie:

```
hide mysql_servers = "localhost/virtualemail/vmail/sekret"
```

Kolejne pola oznaczają: nazwę serwera z bazą (lista kont może być zatem na osobnej maszynie), nazwę bazy danych, użytkownika i hasło. Słowo kluczowe `hide` uniemożliwia pokazanie tej linii przez wywołanie `exim -bP`. Oczywiście trzeba zadbać o odpowiednie prawa dostępu do pliku `exim.conf`, ponieważ hasło podajemy w nim jawnie.

Wszystkie domeny obsługiwane przez system *Vmail-SQL* muszą być dodane do `local_domains`. Zamiast jednak umieszczać je bezpośrednio w pliku konfiguracyjnym można wskazać zewnętrzny plik z listą:

```
local_domains = przyklad.pl:lsearch;/etc/exim/vmail_domains

$ cat /etc/exim/vmail_domains
klient1.pl
drugi.klient.pl
$
```

### 3.3.2 `exim.conf`: Sekcja *transports*

```
virtual_localdelivery:
  driver = appendfile
  file = ${lookup mysql{select path from domain where
         domain_name = '$domain'}{$value}fail}/${lookup
         mysql{select mbox_name from popbox where domain_name =
         '$domain' and local_part = '$local_part'}{$value}fail}
  delivery_date_add
  envelope_to_add
  return_path_add
  user = ${lookup mysql{select unix_user from domain where
         domain_name = '$domain'}{$value}fail}
  mode = 0660
```

Uwaga: wartości opcji `file` i `user` powinny być wpisane w jednym wierszu, lub podzielone znakami „\" (*backslash*).

Transport *virtual_localdelivery* zajmuje się dostarczaniem listu do skrzynki odpowiedniego użytkownika. Miłośnicy SQL'a dostrzegą z łatwością tzw. *select'y* służące wyszukiwaniu informacji w bazie danych.

Kolejność transportów nie ma znaczenia, możemy więc w ramach tej sekcji dopisać powyższy kod w wybranym miejscu.

### 3.3.3 `exim.conf`: Sekcja *directors*

```
virtual_forward:
  domains = lsearch;/etc/exim/vmail_domains
  driver = aliasfile
  search_type = mysql
  query = "select remote_name from forwarder, domain where
           local_part = '$local_part' and forwarder.domain_name =
           domain.domain_name and domain.domain_name = '$domain'"
  forbid_file = true
  forbid_pipe = true

virtual_localuser:
  domains = lsearch;/etc/exim/vmail_domains
  driver = aliasfile
  search_type = mysql
  query = "select mbox_name from popbox, domain where
           local_part = '$local_part' and popbox.domain_name =
           domain.domain_name and domain.domain_name = '$domain'"
  transport = virtual_localdelivery

virtual_defaultuser:
  domains = lsearch;/etc/exim/vmail_domains
  driver = aliasfile
  search_type = mysql
  query = "select remote_name from forwarder where
           local_part = '_default_' and domain_name = '$domain'"
  forbid_file = true
  forbid_pipe = true
```

Choć z pozoru konfiguracja wygląda dość skomplikowanie, po przejrzeniu podręcznika do *Exima* wszystko stanie się jasne :-) Istotny jest parametr `domains`, w którym określamy listę domen związanych z danym *director'em*. Ponieważ zwykle chcemy zachować możliwość wysyłania poczty na standardowe, Uniksowe konta, do pozostałych *director'ów* (np. *system_aliases*, *procmail*, *localuser*) dopisujemy wyrażenie:

```
domains = !lsearch;/etc/exim/vmail_domains
```

Dzięki temu *Exim* będzie używał standardowych procedur dla domen **nie obsługiwanych** przez *Vmail-SQL*.

Jak już wspomniałem, kolejność *director'ów* jest istotna. *Exim* podejmuje decyzję o tym, jak dostarczyć list, próbując uruchamiać je w kolejności wystąpienia w pliku konfiguracyjnym. Jeśli większość e-maili obsługiwanych przez serwer trafia na konta SQL-owe, proponuję umieścić wyżej wymienione dyrektywy na początku sekcji.

## 3.4 Edytujemy `tpop3d.conf`

Opcje dostępne w pliku konfiguracyjnym opisane są dokładnie w `man tpop3d.conf`. Poniżej zamieszczę jedynie przykładowy plik wraz z wyjaśnieniem najważniejszych opcji.

```
# global
listen-address: 10.0.0.1(klient1.pl)
max-children: 30
append-domain: yes
timeout-seconds: 30
mailbox: /var/spool/mail/$(user)

# PAM
auth-pam-enable: yes
auth-pam-facility: tpop3d
auth-pam-mail-group: mail

# MySQL
auth-mysql-enable: yes
auth-mysql-mail-group: mail
auth-mysql-hostname: localhost
auth-mysql-database: virtualemail
auth-mysql-username: vmail
auth-mysql-password: sekret
```

Składnia jest według mnie czytelna; część parametrów ma znaczenie globalne, reszta natomiast (`auth-...`) dotyczy poszczególnych modułów autoryzacji. W powyższym przykładzie skonfigurowałem dwa mechanizmy autoryzacji: poprzez PAM i MySQL.

`listen-address` Wskazuje adres IP, na którym „nasłuchuje" demon POP3. Możemy dzięki temu obsługiwać wiele serwerów wirtualnych posadowionych na jednym komputerze. Oczywiście najpierw

trzeba skonfigurować interfejs sieciowy związany z powyższym adresem IP. W typowym przypadku wystarczy dodać alias do interfejsu, np. *eth0:0* komendą `ifconfig`.

W nawiasie możemy opcjonalnie podać nazwę domeny związanej z danym IP; jeśli mamy dobrze skonfigurowany *reverse DNS* nie jest to wymagane.

`append-domain` Parametr ten uruchamia automatyczne „doklejanie" domeny do nazwy użytkownika. Opcja ma istotne znaczenie, jeśli użytkownicy nie są przyzwyczajeni do wpisywania „luser@klient1.pl" w polu na nazwę użytkownika swojego programu pocztowego.

`mailbox` Określa lokalizację skrzynek pocztowych; w przypadku autoryzacji MySQL parametr jest ignorowany (lokalizacja pobierana z bazy).

Ponieważ hasło do bazy danych podano jawnie, należy szczególnie zabezpieczyć pliki konfiguracyjne przed niepożądanym odczytem.

Po stworzeniu plików konfiguracyjnych uruchamiamy demon POP3:

```
# tpop3d -f /etc/tpop3d/klient1.pl
#
```

Jeśli wszystko jest w porządku, nie zobaczymy żadnych komunikatów, a w tablicy procesów pojawi się `tpop3d`. Logi informujące o różnych zdarzeniach, np. zalogowaniu się użytkownika przesyłane są do *syslog'a*, w sekcji (*facility*) *mail*.

Do celów poznawczych i testowania możemy uruchomić serwer w taki sposób:

```
# tpop3d -f /etc/tpop3d/klient1.pl -v -d
listening on address 10.0.0.1, port 110, domain klient1.pl
2 authentication drivers successfully loaded
net_loop: tpop3d version 1.3.2 successfully started
...
```

Każda operacja związana z serwerem będzie szczegółowo logowana dzięki opcji `-v`, natomiast opcja `-d` pozostawi *tpop3d* jako proces pierwszoplanowy na terminalu.

# 4 Użytkowanie

Ufff, po tylu ceregielach mamy wreszcie serwer przygotowany na przyjęcie pierwszych użytkowników. Pierwsze pytanie jakie zapewne zostanie zadane to: *jak w tym zamieszanym systemie zakładać konta?* Otóż, istnieją co najmniej trzy sposoby:

1. ręczne dopisywanie kont przy użyciu SQL-owych komend,
2. automatyzacja pierwszego sposobu z pomocą własnych skryptów (np. w Perlu czy PHP),
3. skorzystanie z gotowych skryptów załączonych w pakiecie *Vmail-SQL* napisanych w Perlu.

Pierwszego sposobu nie polecam ze względu na czasochłonność, drugi jest niezły dla zaawansowanych programistów i adminów, ale na początek opiszmy ten trzeci :-)

W podkatalogu *vmail-sql-0.3/scripts/* mamy cztery skrypty:

**VE-domain** służący do zakładania nowej domeny,

**VE-popbox** do tworzenia kont użytkowników,

**VE-forwarder** do zakładania aliasów (przekierowań),

**VE-passwd** do zmiany haseł domeny i użytkownika.

W każdym ze skryptów musimy wskazać namiary na naszą bazę danych. Szukamy więc miejsca „INSERT YOUR DSN HERE" i wprowadzamy np.:

```
$dsn = "DBI:mysql:virtualemail;localhost";
$dbh = DBI->connect($dsn, 'vmail', 'sekret'); # INSERT YOUR DSN HERE
```

Po raz trzeci (i pewnie nie ostatni) wpisujemy jawnie hasło, więc obowiązuje stara zasada maksymalnego zabezpieczania skryptów przez odczytem.

Sposób korzystania z programów powinien być intuicyjny, po uruchomieniu skryptu bez parametrów dostaniemy krótką instrukcję. Pozwolę sobie zamieścić tylko skrótowo sesję z tworzenia nowej domeny i zakładania konta.

```
# VE-domain add klient1.pl
Will add domain ‘klient1.pl’ using Unix user vmail
Used directory ‘/var/spool/mail/SERVERS/klient1.pl’
Done
# VE-domain popboxes klient1.pl max 50
Done.
# VE-passwd klient1.pl haslo_domeny
Done.
# VE-popbox add klient1.pl luser
Will add POP3 maildrop for ‘luser@klient1.pl’ with maildrop ‘luser’
Done
# VE-passwd luser@klient1.pl haslo_lusera
Done
```

Jeśli wszystko po drodze ustawiliśmy dobrze to powinny zadziałać następujące testy:

**Test 1:** *Czy działa SMTP?*

```
# echo test | mail luser@klient1.pl
# ls -l /var/spool/mail/SERVERS/klient1.pl/luser
-rw-rw----    1 vmail    nogroup       454 Jun  5 19:06 luser
```

W skrzynce użytkownika pojawił się nasz list. Wniosek: SMTP działa!

**Test 2:** *Czy działa POP3?*

```
# telnet 10.0.0.1 110
Trying 10.0.0.1...
Connected to 10.0.0.1.
Escape character is ’^]’.
+OK <ffe36b8dae9197aacb32a1ec5021c78b@klient1.pl>
user luser@klient1.pl
+OK Tell me your password.
pass haslo_lusera
+OK Welcome aboard! You have 1 messages.
quit
+OK Done
Connection closed by foreign host.
```

Serwer przeprowadził poprawnie autoryzację i otworzył skrzynkę pocztową. Wniosek: POP3 działa!

W pakiecie *Vmail-SQL* znajdziemy jeszcze jeden katalog: `web`, którego nazwa nieodmiennie wskazuje powiązanie z WWW. Tę część pakietu możemy oczywiście pominąć, jeśli interesuje nas jedynie zakładanie kont z *shell'a*, ale zwykle przydałaby się użytkownikom możliwość zmiany hasła. Nie ma przeszkód aby napisać do tego celu własny interfejs w dowolnym języku obsługującym MySQL, ale i tym razem pójdźmy na łatwiznę używając gotowca.

Obsługa systemu przez WWW zawiera kilka komponentów i umożliwia nie tylko zmienianie hasła użytkowników, ale również pełne zarządzanie kontami. Możemy dzięki temu udostępnić interfejs swojemu klientowi i zaoszczędzić trochę czasu.

Aby tradycji stało się zadość musimy najpierw skonfigurować dostęp do bazy edytując plik `lib/DomainAdminConfig.pm`. Zawartość katalogu `docs/` powinna trafić do katalogu, gdzie normalnie umieszczamy strony. `cgi-bin/*` jak nazwa wskazuje wrzucamy do katalogu ze skryptami CGI, natomiast pliki z `lib/*` pod żadnym pozorem nie powinny być widoczne z poziomu przeglądarki WWW (umieściliśmy w nim hasło dostępu!).

Być może będzie konieczne lekkie „przemeblowanie" struktury katalogów, lecz dla średnio zaawansowanego administratora nie powinno to sprawić problemów. Należy pamiętać o podaniu ścieżek do bibliotek dołączanych w skryptach CGI.

Po skonfigurowaniu interfejsu web'owego kierujemy przeglądarkę na adres: http://www.przyklad.pl/cgi-bin/login. Do zalogowania się potrzebne jest hasło ustalone przy zakładaniu domeny. Można je zmienić programem `VE-passwd`. Przez WWW możemy od tego momentu zakładać/kasować/modyfikować konta i przekierowania. Otwartość kodu ułatwia wszelkie modyfikacje i dostrajanie systemu do własnych potrzeb. Dla przykładu: zajęło mi około dwóch godzin przerobienie wyglądu stron i przetłumaczenie komunikatów na język polski.

W praktyce może się jednak okazać, że nie wszystko działa tak od ręki jak w powyższych przykładach. Zbyt wiele jest zmiennych czynników po drodze aby opisać w niniejszym artykule wszystkie możliwości. W razie kłopotów mogę jedynie polecić dokładną obserwację komunikatów i logów systemowych. Warto też przestudiować dokumentację dołączoną do pakietu *Vmail-SQL* oraz *Exim'a*. Jeśli nadal jesteśmy w kropce, pozostaje skontaktowanie się z jakimś ekspertem :-) Można takich znaleźć np. na liście dyskusyjnej *vmail-discuss*: http://lists.beasts.org/mailman/listinfo/vmail-discuss

# 5 Co dalej?

## 5.1 Wady *Vmail-SQL*

Oczywiście takowe istnieją... Moim zdaniem nie są to duże niedociągnięcia, ale jednak byłbym nierzetelny nie wspominając o nich.

**Kod we wczesnej fazie rozwoju.** Ze względu na niewielką liczbę użytkowników, kod nie jest w pełni przetestowany.

**Brak obsługi limitów dyskowych (*quotas*).** Na poziomie systemu operacyjnego można przydzielać różne *quoty* użytkownikom, jednak należy pamiętać, że *Vmail-SQL* zapisuje pocztę dla całej domeny z prawami jednego użytkownika Uniksowego. Dlatego nie można precyzyjnie w ramach domeny kontrolować przydziału miejsca. Problem ten jest łatwy do rozwiązania: *Exim* obsługuje limity niezależnie od systemu operacyjnego. Wystarczy w pliku `exim.conf` wpisać w sekcji transportu `virtual_localdelivery` linię:

```
quota = 20M
```

Dwadzieścia megabajtów to oczywiście przykład. Można pokusić się o modyfikację bazy danych i dodanie pola *quota* do tabeli *popbox*. *Exim* mógłby wówczas dynamicznie przydzielać różne limity użytkownikom.

**Niedopracowane narzędzia do obsługi przez WWW.** Wymagają one obecnie dostrojenia do potrzeb użytkownika, przypuszczam jednak, że wraz ze wzrostem popularności oprogramowanie będzie rozwijane.

**Wsparcie tylko dla języka angielskiego.** Współpracując z autorem stworzyłem polskie tłumaczenie serwera `tpop3d`, które niedługo być może znajdzie się w oficjalnej dystrybucji.

## 5.2 Alternatywne rozwiązania

*Vmail-SQL*, jak nietrudno się domyśleć, nie jest jedynym pakietem obsługujących pocztę dla wielu serwerów wirtualnych. Trzeba powiedzieć, że zajmuje raczej jedno z ostatnich miejsc według popularności. Może to właśnie dlatego przykuł moją uwagę :-)

Poniżej opisano alternatywne pakiety wygrzebane z przepastnych zasobów Internetu.

**Courier, Courier-IMAP** Zintegrowany zestaw narzędzi do obsługi SMTP, POP3, IMAP, webmail itp.

Źródło: http://courier.sourceforge.net/,
http://www.inter7.com/courierimap/.

**exim-qpopper-mysql** Patch do *Qpopper'a* i pliki konfiguracyjne *Exim'a* dodające obsługę MySQL. Dostępne narzędzia w PHP do obsługi systemu przez WWW.

Źródło: http://www.netd.co.za/mysql-mail/

**ISPMan** Kompletny system zarządzania dla ISP z poziomu przeglądarki WWW – m.in. obsługuje e-mail przy użyciu *Postfix'a* jako serwera SMTP i *Cyrus'a* do IMAP/POP3.

Źródło: http://www.ispman.org/

**mmmail** Proste serwery SMTP i POP3 działające bez praw root'a. Lista użytkowników przechowywana w bazie MySQL. Trochę zbyt ograniczone możliwości.

Źródło: http://mmondor.rubiks.net/

**pop3lite** Interesujący projekt: sam serwer obsługuje minimalny zestaw funkcji, ale jest łatwo rozszerzalny o moduły pełniące różne zadania, np. obsługę wirtualnych domen.

Źródło: http://pop3lite.sourceforge.net/

**POPular** Serwer dedykowany dla dużych ISP, serwujących co najmniej kilka tysięcy kont. Autoryzacja w oparciu o zewnętrzne moduły, aktualnie dostępne: Berkeley DB, CDB.

Źródło: http://www.remote.org/jochen/mail/popular/

**SolidPOP** Polska produkcja, obsługuje zarówno domeny wirtualne na wielu jak i na jednym IP. Wymaga mapowania użytkowników „wirtualnych" na rzeczywiste konta Uniksowe.

Źródło: http://solidpop3d.pld.org.pl/

**Virtuald** Demon do wirtualizacji różnych serwisów sieciowych, m.in. SMTP (Sendmail, Qmail) i POP3 (Qpopper). Dedykowany dla RedHat'a.

Źródło: http://www.linuxdoc.org/HOWTO/Virtual-Services-HOWTO.html

**virtualmail-pop3d, vm-pop3d** Podobnie jak *tpop3d* jest to niewielki, ale funkcjonalny serwer POP3, jednak zamiast bazy danych stosowany jest prosty plik tekstowy, podobny do `/etc/passwd`. Obsługuje domeny wirtualne, zarówno na jednym, jak i na wielu IP. Jako serwer SMTP preferowany jest *Exim* lub *Postfix*.

Źródło: http://www.reedmedia.net/software/virtualmail-pop3d/

**VMailMgr** Jeden z najbardziej popularnych pakietów, zwłaszcza wśród zwolenników *Qmail'a*. Baza użytkowników przechowywana w CDB, czyli wydajnej bazie D.J. Bernsteina.

Źródło: http://www.vmailmgr.org/

**vpopmail** Znany pakiet, dedykowany dla serwisów z milionami kont pocztowych. Współpracuje z *Qmail'em* i *Postfix'em*, autoryzacja poprzez moduły; aktualnie dostępne: `/etc/passwd`, CDB, LDAP, MySQL, Oracle, Sybase.

Źródło: http://www.inter7.com/vpopmail/