Postfix Debian Etch 4.0 + dodatki

Autor: Wiktor Łyczywek
ver. 1.1
wiktor.lyczywek(at)gmail(dot)com

Spis treści:

1. Wstęp

1.1 Postfix

Postfix to szybki, łatwy w administracji (w porównaniu z innymi dostępnymi aplikacjami tego typu) i bezpieczny zestaw oprogramowania umożliwiającego stworzenie serwera pocztowego. Zbudowany jest on z wielu mniejszych programów współpracujących ze sobą. Pierwsza zaleta (patrząc od strony bezpieczeństwa) jest fakt, i kolejki FIFO wykorzystywane do komunikacji pomiędzy owymi programami znajdują się w chronionych katalogach. Podstawowymi czterema kolejkami systemu Postfix sa:

- *maildrop* – kolejka przechowująca pocztę wysyłana lokalnie, która następnie trafi z tej kolejki do incoming;
- *incoming* – kolejka poczty przychodzącej oraz nie przetworzonej jeszcze przez program wchodzący w skład Postfix – Queue Manager;
- *active* – zawiera pocztę przetworzona przez Queue Manager i gotowa do dostarczenia;
- *deferred* – kolejka przechowująca pocztę, której nie udało się dostarczyć.

Postfix posiada także mechanizm, gwarantujący płynne przetwarzanie maili. Po pierwsze, *Queue Manager* działa w taki sposób, że do przetwarzania pobiera na przemian po jednej wiadomości z kolejki *incoming* oraz z *deferred*. Po drugie, Postfix stara się nie nawiązywać więcej niż dwóch połączeń, gdy dostarcza pocztę do konkretnej lokalizacji, dopóki kolejne wysyłane przesyłki nie będą dostarczane bezproblemowo. Dodatkowo maile, których nie udało się wysłać są oznaczane specjalnym znacznikiem czasowym, który wraz z kolejnymi nieudanymi próbami jest powiększany dwukrotnie, co blokuje jego przetwarzanie przez *Queue Manager*. Kolejne zabezpieczenie Postfix polega na zapamiętywaniu hostów, do których nie udało się dostarczyć przesyłki, aby w przyszłości uniknąć próby wysyłania do nich.

1.2 SMTP (Simple Mail Transfer Protocol)

SMTP (ang. Simple Mail Transfer Protocol) - protokół komunikacyjny opisujący sposób przekazywania poczty elektronicznej w Internecie.

SMTP to względnie prosty, tekstowy protokół, w którym określa się co najmniej jednego odbiorcę wiadomości (w większości przypadków weryfikowane jest jego istnienie), a następnie przekazuje treść wiadomości. Demon SMTP działa najczęściej na porcie 25. Łatwo przetestować serwer SMTP przy użyciu programu telnet.

SMTP nie pozwala na pobieranie wiadomości ze zdalnego serwera. Do tego celu służą POP3 lub IMAP.

Jednym z ograniczeń pierwotnego SMTP jest brak mechanizmu weryfikacji nadawcy, co ułatwia rozpowszechnianie niepożądanych treści poprzez pocztę elektroniczną (wirusy, spam). Żeby temu zaradzić stworzono rozszerzenie SMTP-AUTH, które jednak jest tylko częściowym rozwiązaniem problemu - ogranicza wykorzystanie serwera wymagającego autoryzacji do zwielokrotniania poczty. Nadal nie istnieje metoda, dzięki której odbiorca autoryzowałby nadawcę - nadawca może "udawać" serwer i wysłać dowolny komunikat do dowolnego odbiorcy.

1.3 POP3 (Post Office Protocol version 3)

POP3 (ang. Post Office Protocol version 3) to protokół internetowy z warstwy aplikacji pozwalający na odbiór poczty elektronicznej ze zdalnego serwera do lokalnego komputera poprzez połączenie TCP/IP. Ogromna większość współczesnych internautów korzysta z POP3 do odbioru poczty

Protokół POP3 powstał dla użytkowników, którzy nie są cały czas obecni w Internecie. Jeżeli ktoś łączy się z siecią tylko na chwilę, to poczta nie może dotrzeć do niego protokołem SMTP. W takiej sytuacji w sieci istnieje specjalny serwer, który przez SMTP odbiera przychodzącą pocztę i ustawia ją w kolejce.

Kiedy użytkownik połączy się z siecią, to korzystając z POP3 może pobrać czekające na niego listy do lokalnego komputera. Jednak protokół ten ma wiele ograniczeń:

- połączenie trwa tylko, jeżeli użytkownik pobiera pocztę i nie może pozostać uśpione,
- do jednej skrzynki może podłączyć się tylko jeden klient równocześnie,
- każdy list musi być pobierany razem z załącznikami i żadnej jego części nie można w łatwy sposób pominąć - istnieje co prawda komenda **top**, ale pozwala ona jedynie określić przesyłaną liczbę linii od początku wiadomości,
- wszystkie odbierane listy trafiają do jednej skrzynki, nie da się utworzyć ich kilku,
- serwer POP3 nie potrafi sam przeszukiwać czekających w kolejce listów.

Istnieje bardziej zaawansowany protokół IMAP, który pozwala na przeglądanie czekających listów nie po kolei na podobieństwo plików w katalogach i posiada niektóre funkcje pominięte w POP3.

Programy odbierające pocztę najczęściej obsługują oba protokoły, ale POP3 jest bardziej popularny. Wysyłanie listów zawsze opiera się na protokole SMTP. Komunikacja POP3 może zostać zaszyfrowana z wykorzystaniem protokołu SSL. Jest to o tyle istotne, że w POP3 hasło przesyłane jest otwartym tekstem, o ile nie korzysta się z opcjonalnej komendy protokołu POP3, APOP.

1.4 IMAP (Internet Message Access Protocol)

IMAP (ang. Internet Message Access Protocol) to internetowy protokół pocztowy zaprojektowany jako następca POP3. W przeciwieństwie do POP3, który umożliwia jedynie pobieranie i kasowanie poczty, IMAP pozwala na zarządzanie wieloma folderami pocztowymi oraz pobieranie i operowanie na listach znajdujących się na zdalnym serwerze.

Powodem, dla którego IMAP się nie przyjął, z pewnością NIE JEST czas połączenia z serwerem. W rzeczywistości protokół ten jest polecany dla użytkowników modemów, którzy nie muszą przez godzinę ściągać całej poczty po to tylko, by się przekonać, że większość maili ich nie interesuje lub zawiera ogromne załączniki. IMAP pozwala na ściągnięcie nagłówków wiadomości i wybranie, które z wiadomości chcemy ściągnąć na komputer lokalny. Zdecydowanie zmniejsza to czas połączenia oraz eliminuje konieczność wchodzenia bezpośrednio na stronę w celu usunięcia wiadomości o zbyt dużym rozmiarze.

Protokół IMAP okazał się jednak nieco skomplikowany, wymaga stałego dostępu do sieci i prawdopodobnie dlatego nie przyjął się powszechnie. Pozwala na wykonywanie wielu operacji, zarządzanie folderami i wiadomościami. Dodatkowo, mało który portal udostępnia darmowe skrzynki obsługiwane przez IMAP, zapewne dlatego, że użytkownik nie byłby zmuszony do pobierania wysyłanego przez portale materiału reklamowego.

1.5 Szyfrowanie i autoryzacja

Autoryzacja poczty wychodzącej (SMTP AUTH) to zabezpieczenie uniemożliwiające wysyłkę poczty bez uprzedniego zalogowania się do konta. To zabezpieczenie uniemożliwia wysyłkę listów email bez uprzedniej autoryzacji użytkownika. Co to znaczy? Dzięki autoryzacji nie możliwe jest podszycie się pod cudzy adres e-mail, a odbiorca wiadomości ma pewność ze adresat jest tym, za kogo się podaje. Innymi słowy, jeśli serwer poczty nie posiada autoryzacji każdy mógłby połączyć się z nim, przykładowo przez telnet i podszywając sie pod różnych właścicieli kont e-mail wysyłać dowolna ilość maili, wykorzystując konto na przykład do rozsyłania spamu.

W każdym unixowym systemie domyślnie instalowany jest jakiś serwer poczty, zazwyczaj jest to sendmail czy te exim. Sa jednak bardziej wydajne i bardziej bezpieczne demony SMTP. Do takich należy dobrze znany Postfix. Jednak zle skonfigurowany może stanowić poważne zagrożenie bezpieczeństwa w sieci. Tak wiec przed instalacja Postfix'a warto pomyslec o tym, aby nasz serwer pocztowy nie stał się od razu po zainstalowaniu open-relay'em. Aby tego uniknąć na samym początku należy zainstalować Cyrusa czyli przygotować SASL'a dla Postifx'a. Jak się łatwo domyślić **SASL (Simple Authentication and Security Layer)** to metoda uwierzytelniania użytkownika (autoryzacja).

Sama autoryzacja SMTP nie zabezpiecza w pełni jeszcze serwera pocztowego. Połączenie miedzy klientem a serwerem jest nieszyfrowane, wiec możliwe jest przechwycenie hasła. Aby zabezpieczyć sie przed taka sytuacja stosuje się szyfrowanie za pomocą protokołu **TLS (Transport Layer Security)**, który jest rozszerzeniem protokołu **SSL (Secure Sockets Layer).** Dzięki TLS wszystkie wiadomości przesyłane i odbierane z danego programu pocztowego będą szyfrowane, co uniemożliwi dostęp do nich niepowołanym osobom. Jednak e ani TLS ani SSL nie chroni przeciw analizie ruchu w sieci. Analizując ruch można ustalić ilość wysłanych wiadomości oraz adresy obu połączeń (przesyłane dane są bezpieczne). Tak, wiec aby autoryzacja miała jakikolwiek sens konieczne jest zaszyfrowanie połączenia miedzy klientem a serwerem.

1.6 SpamAssassin

SpamAssassin - napisany głównie w perlu zestaw skryptów do skanowania zawartości poczty elektronicznej i oceny prawdopodobieństwa czy dana wiadomość jest spamem, czy też nie.

W swoich oznaczeniach program stosuje metodę punktową, gdzie im wyższa ocena, tym większe prawdopodobieństwo że treść wiadomości jest niepożądana. Posiada możliwość "uczenia się" wiadomości chcianych i nie chcianych co przy zastosowaniu odpowiedniej ilości przykładowych wiadomości pozwala na osiągnięcie całkiem zadowalających efektów filtracji.

1.7 Clam AntiVirus (ClamAV)

Clam AntiVirus (ClamAV) - zestaw narzędzi antywirusowych, dostępnych na licencji GPL działający pod systemami Uniksowymi.

ClamAV jest przeznaczony głównie do integracji z serwerami pocztowymi (skanowanie załączników). W skład zestawu wchodzą między innymi:

- wielowątkowy demon
- skaner
- narzędzie do tworzenia sygnatur i własnych baz wirusów
- biblioteka, dzięki której można tworzyć własne programy antywirusowe na bazie ClamAV
- narzędzie do automatycznej i darmowej aktualizacji bazy wirusów, baza zawierająca sygnatury ponad 90.000 wirusów

Istnieje też jego odpowiednik ClamWin działający w systemach Windows oraz KlamAV integrujący się ze środowiskiem graficznym KDE.

ClamAV to nie tylko program antywirusowy, to także, cała gama programów mu towarzyszących. Stanowią one połączenie między ClamAV a usługami takimi jak: MTA, MUA, serwer POP3, serwer proxy, tworzą graficzny interfejs dla ClamAV, czy wreszcie wzbogacają system o taką funkcjonalność jak skanowanie, przy dostępie (do pliku).

2. Postfix

2.1 Instalacja

Instalacja pakietów podstawowych, niezbędnych do uruchomienia serwera Postfix:

- *postfix* – agent poczty,
- *postfix-doc* – dokumentacja dla postfix'a,
- *openssl* - biblioteka funkcji kryptograficznych i obsługi certyfikatów.

Shell

```
# aptitude install postfix postfix-doc openssl
```

Gdy konfigurator przejdzie do postfixa "Klikamy OK", Następnie wybieramy „No configuration" i OK.

2.2 Wstępna konfiguracja

Tworzymy lub edytujemy plik konfiguracyjny postfixa:

Shell

```
# nano /etc/postfix/main.cf
```

Wklejamy i edytujemy zawartość poniższego konfiga

File

```
command_directory = /usr/sbin
mail_owner = postfix
mydomain = moja-domena.pl
myhostname = moja-domena.pl

myorigin = /etc/mailname
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases

mydestination = $mydomain, $myhostname, localhost, localhost.localdomain,

mynetworks = 127.0.0.0/8   #[10.0.0.0/24] - Twoja siec
inet_interfaces = all
inet_protocols = ipv4

home_mailbox = Maildir/
mail_spool_directory = /home/

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)

mailbox_size_limit = 0
recipient_delimiter = +

relayhost =
```

A teraz wyjaśnienie opcji:

command_directory	katalog lokalizujący demona
mail_owner	użytkownik pod którym Postfix wykonuje większość operacji
mydomain	nazwa domeny
myhostname	nazwa hosta (tak przedstawia się serwer EHLO)
myorigin	co będzie dodawane po @, gdy brak domeny w adresie z pola from myorigin = $myhostname ("user@$myhostname") myorigin = $mydomain ("user@$mydomain")
alias_maps alias_database	stosowane bazy aliasów
mydestination	jakie domeny docelowe (poza wirtualnymi) maja być akceptowane myorigin = $myhostname ("user@$myhostname") myorigin = $mydomain ("user@$mydomain")
mynetworks mynetworks_style	określenie komputerów, których pocztę będziemy przekazywać mynetworks_style = subnet (z klientów SMTP z naszej podsieci) mynetworks_style = host (tylko z maszyny lokalnej) mynetworks = 127.0.0.0/8 (tylko lokalna maszyna) mynetworks = 127.0.0.0/8 168.100.189.2/32
inet_interfaces	adres interfejsu, na którym będzie nasłuchiwał postfix inet_interfaces = $myhostname, localhost (nasz host oraz localhost) inet_interfaces = all (wszystkie interfejsy)
inet_protocols	wersja protokołu
home_mailbox	typ skrzynek
mail_spool_directory	katalog z poczta dla skrzynek mailbox
smtpd_banner	sposób przedstawiania się przez serwer
mailbox_size_limit	maksymalny rozmiar skrzynki pocztowej (0 . brak limitu)

recipient_delimiter	niezbędne do monitorowania domen wirtualnych
relayhost =	…
message_size_limit	Maksymalna wielkość wysyłanej wiadomości message_size_limit = 10240000 (10MB)
queue_run_delay maximal_queue_liftime	Parametry odpowiedzialne za dostarczenie maila jeżeli serwer jest wyłączony (queue_run_delay = 30m maximal_queue_liftime = 3d sprawdzaj zdalny Server co pół godziny i zwracaj błąd do nadawcy po trzech dniach)
unknown_local_recipient_reject_code	Numer kodu, który będzie zwracany, gdy w systemie nie ma odbiorcy unknown_local_recipient_reject_code = 550
local_destination_concurrency_limit	Ustawia limit ilości wiadomości, które mogą być dostarczone do użytkownika w tym samym czasie
default_destination_concurrency_limit	Ustawia limit ilości użytkowników, których Postfix może obsłużyć równocześnie w tym samym czasie

Tworzymy bazę aliasów:

```
Shell
# newaliases
```

Sprawdzamy składnie plików konfiguracyjnych:

```
Shell
# postfix check
```

Restartujemy daemona:

```
Shell
# /etc/init.d/postfix restart
```

2.3 Test

W tym momencie możemy już sprawdzić, czy nasz deamon SMTP działa. Dokonujemy tego z wykorzystaniem komendy telnet:

Shell

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 moja-domena.pl ESMTP Postfix
EHLO moja-domena.pl
250-moja-adomena.pl
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
quit
221 Bye
Connection closed by foreign host.
```

Powinniśmy zobaczyć komunikat jak powyżej mniej więcej.

3. Serwer POP3 i IMAP

Postfix jest agentem MTA służącym do nasłuchiwania na porcie 25 i ewentualnie przekazywania odebranej poczty dalej. Aby mieć równie możliwość odbierania poczty, dokonamy instalacji serwera POP3 i IMAP. Wykorzystamy do tego zestaw pakietów decovet:

3.1 Instalacja

- *dovecot-common* – dokumentacja dla serwerów POP i IMAP
- *dovecot-imapd* - serwer IMAP wspierający mbox i maildir poczty
- *dovecot-pop3d* - serwer POP3 wspierający mbox i maildir poczty

```
Shell
# aptitude install dovecot-imapd dovecot-pop3d dovecot-common
```

Ustawmy prawa dla katalogów:

```
Shell
# chmod 755 /var/run/dovecot
# chgrp dovecot /var/run/dovecot/login/
```

Tworzymy katalog

```
Shell
# mkdir -p /etc/dovecot/ssl
```

Przechodzimy do katalogu

```
Shell
# cd /etc/dovecot/ssl
```

Tworzymy certyfikat

```
Shell
# openssl req -new -x509 -nodes -out dovecot.pem -keyout dovecot.pem -days 365
```

Przy tworzeniu certyfikatu wprowadzamy potrzebne dane:

Shell

```
Country Name (2 letter code) [AU]:PL
State or Province Name (full name) [Some-State]:POLAND
Locality Name (eg, city) []:Warszawa
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Moj certifikat
Organizational Unit Name (eg, section) []:POCZTA
Common Name (eg, YOUR name) []:moja-domena.pl
Email Address []:root@moja-domena.pl
```

UWAGA!!! Ważne żeby w polu **Common Name** podać własną domenę

3.2 Konfiguracja

Edytujemy plik:

Shell

```
# nano /etc/dovecot/dovecot.conf
```

I wklejamy poniższy konfig

File

```
protocols = imap imaps pop3 pop3s
listen = *
log_timestamp = "%Y-%m-%d %H:%M:%S "
log_path=/var/log/dovecot.log

login_process_size = 64
login_greeting = POP ready
mail_location = maildir:~/Maildir

#SSL
ssl_disable = no
ssl_cert_file = /etc/dovecot/ssl/dovecot.pem
ssl_key_file = /etc/dovecot/ssl/dovecot.pem
verbose_ssl = yes

#namespace private {
#  separator = .
#  prefix = INBOX.
#  inbox = yes
#  hidden = yes
#}

#namespace private {
#  separator = .
#  prefix =
#  inbox = yes
#}

mail_extra_groups = postfix
```

```
# jesli wszystko bedzie dzialac polecam wylaczyc debug
auth_debug = yes
auth_verbose = yes
verbose_proctitle = yes

protocol imap {
}

protocol pop3 {
        pop3_uidl_format = %08Xu%08Xv
}

protocol lda {
        postmaster_address = postmaster@moja-domena.pl
}

auth default {
  mechanisms = plain

  passdb pam {
  }
  userdb passwd {
  }
  user = root
}

dict {
}

plugin {
}
```

Startujemy daemona

Shell

```
# /etc/init.d/dovecot start
```

Wszystko powinno być OK, ale jeżeli dovecot nie chce wystartować należy sprawdzić logi.

3.3 Test

Testowanie serwera SMTP (w systemie musi istnieć konto nadawca i adresat):

Shell

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
220 moja-domena.pl ESMTP Postfix
EHLO moja-domena.pl
250-moja-domena.pl
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-ENHANCEDSTATUSCODES
```

```
250-8BITMIME
250 DSN
mail from: nadawca@moja-domena.pl
250 Ok
rcpt to: adresat@moja-domena.pl
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: mail testowy
bla
bla bla
.
250 2.0.0 Ok: queued as DF6693AE2F3
quit
221 Bye
Connection closed by foreign host.
```

Testowanie serwera POP3 – nieszyfrowanego:

Shell

```
# telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK Dovecot ready.
user adresat
+OK
pass haslo
+OK Logged in.
stat
+OK 1 513
list
+OK 1 messages:
1 513
.
retr 1
+OK 513 octets
Return-Path: <nadawca@moja-domena.pl>
X-Original-To: adresat@moja-domena.pl
Delivered-To: adresat@moja-domena.pl
Received: from moja-domena.pl(localhost [127.0.0.1])
        by palacyk.no-ip.org (Postfix) with ESMTP id DF6693AE2F3
        for <adresat@moja-domena.pl>; Mon,  5 Mar 2007 23:34:53 +0100 (CET)
Subject: mail testowy
Message-Id: <20070305223521.DF6693AE2F3@moja-domena.pl>
Date: Mon,  5 Mar 2007 23:34:53 +0100 (CET)
From: nadawca@moja-domena.pl
To: undisclosed-recipients:;

bla
bla bla
.
quit
+OK Logging out.
Connection closed by foreign host.
```

Testowanie serwera POP3 – szyfrowanego:

Shell

```
# openssl s_client -connect localhost:995

<tutaj pojawią się informacje dotycząca klucza>

+OK Dovecot ready.
user odbiorca
+OK
pass haslo
+OK Logged in.

stat
+OK 1 513

retr 1
+OK 513 octets
Return-Path: <nadawca@moja-domena.pl>
X-Original-To: odbiorca@moja-domena.pl
Delivered-To: odbiorca@moja-domena.pl
Received: from moja-domena.pl(localhost [127.0.0.1])
        by moja-domena.pl (Postfix) with ESMTP id DF6693AE2F3
        for < odbiorca@moja-domena.pl>; Mon,  5 Mar 2007 23:34:53 +0100 (CET)
Subject: mail testowy
Message-Id: <20070305223521.DF6693AE2F3@moja-domena.pl>
Date: Mon,  5 Mar 2007 23:34:53 +0100 (CET)
From: nadawca@moja-domena.pl
To: undisclosed-recipients:;

bla
bla bla
.
quit+OK Bye-bye.
closed
```

4. SASL czyli autoryzacja SMTP

Kolejnym etapem naszych prac, będzie instalacja pakietu SASL, do autoryzacji poczty wychodzącej SMTP, co zapewni nam nieautoryzowane wykorzystywanie naszego konta pocztowego.

4.1 Instalacja

Instalujemy pakiety:

- *libsasl2* – identyfikacja i autoryzacja u użytkownika,
- *sasl2-bin* – program do zarządzania baza u użytkowników,
- *libsasl2-modules* – autoryzacja wirtualnych maili w bazie.

Shell

```
# aptitude install libsasl2 sasl2-bin libsasl2-modules
```

4.2 Konfiguracja

Edytujemy plik:

Shell

```
# nano /etc/postfix/main.cf
```

I dodajemy poniższa zawartość:

File

```
#SASL
smtpd_sasl_auth_enable = yes
smtpd_sasl2_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_sasl_local_domain = $myhostname

smtpd_recipient_restrictions =
      permit_sasl_authenticated
      permit_mynetworks
      reject_unauth_destination
```

I wyjaśnienie użytych opcji:

smtpd_sasl_auth_enable smtpd_sasl2_auth_enable	włączenie autoryzacji SASL
smtpd_sasl_security_options	nie przyjmuje podczas autoryzacji opcji anonymous

broken_sasl_auth_clients	zgodność ze starszymi klientami (Outlook Express 4)
smtpd_sasl_local_domain	autoryzacja wymagana przy wysyłaniu mail z …
smtp_recipient_restrictions	restrykcje na podstawie adresu odbiorcy
permit_sasl_authenticated	dopuść połączenia z autoryzacja
permit_mynetworks	dopuść połączenia z mojej sieci
reject_unauth_destination	odrzuć jeśli nie jest do naszego serwera

Teraz aktywujemy saslauthd'a:

Shell

```
# nano /etc/default/saslauthd
```

i edytujemy linie i ustawiamy jak poniżej:

File

```
START=yes
MECHANISMS="pam"
```

Nie jest to wszystko - drugi plik, którym musimy się zająć (a wręcz stworzyć), to */etc/postfix/sasl/smtpd.conf*. Opisuje on jakim mechanizmem posłużymy się do autentykacji - możemy użyć baz danych lub też w opisywanym przypadku, skonfigurowanego przed chwilą demona odwołującego się do kont systemowych.

Shell

```
# nano /etc/postfix/sasl/smtpd.conf
```

i wklejamy:

File

```
pwcheck_method: saslauthd
mech_list: plain login
```

W perfekcyjnym świecie to byłoby wszystko, w tym mniej doskonałym okazuje się, że dostaniemy komunikat

File

```
postfix/smtpd[7663]: warning: SASL authentication failure: cannot connect to
saslauthd server: No such file or directory
postfix/smtpd[7663]: warning: SASL authentication failure: Password verification
failed
postfix/smtpd[7663]: warning: SASL PLAIN authentication failed
```

Dzieje się tak, ponieważ domyślnie sasl tworzy swój socket w */var/run/saslauthd/* a postfix, działający w chrootowanym środowisku szuka plików w (domyślnie) */var/spool/postfix/var/run/saslauthd/*. Tworzenie katalogu (*mkdir -p /var/spool/postfix/var/run/saslauthd*) a potem tworzenie softlinka (ln -s) jest tu oczywiście opcję, niemniej nie rozwiązującą sytuacji. Dużo lepszym rozwiązaniem (i dość stałym), jest zamontowanie tego katalogu poprzez dowiązanie. Na stałe rozwiązuje to odpowiednia linijka w */etc/fstab*.
Metoda zmuszenia postfixa do pracy z saslauthd.

Włączamy sasla

Shell

```
# /etc/init.d/saslauthd start
```

Dodajemy postfixa do grupy sasl

Shell

```
# adduser postfix sasl
```

Tworzymy katalog dla sasla

Shell

```
# mkdir -p /var/spool/postfix/var/run/saslauthd
```

Montujemy

Shell

```
# mount -o bind /var/run/saslauthd /var/spool/postfix/var/run/saslauthd/
```

Edytujemy /etc/fstab

Shell

```
# nano /etc/fstab
```

I dodajemy wpis

File

```
/var/run/saslauthd /var/spool/postfix/var/run/saslauthd auto bind 0 0
```

Restartujemy saslauthd'a i postfixa

Shell

```
# /etc/init.d/saslauthd restart
# /etc/init.d/postfix restart
```

4.3 Test SASL

Sprawdzamy czy działa

Shell

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 moja-domena.pl ESMTP Postfix
EHLO moja-domena.pl
250-moja-domena.pl
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

Jeżeli są te dwie linijki które są podkreślone to znaczy że wszystko działa.

4.4 Szyfrowanie TLS

Znów edytujemy

Shell

```
# nano /etc/postfix/main.cf
```

I dodajemy poniższe wartości do pliku:

File

```
#TLS
smtpd_tls_auth_only = yes
smtp_use_tls = yes
smtpd_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_tls_key_file = /etc/postfix/ssl/smtpd.pem
smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.pem
smtpd_tls_CAfile = /etc/postfix/ssl/smtpd.pem
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
smtpd_tls_session_cache_database = btree:${queue_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${queue_directory}/smtp_scache
tls_random_source = dev:/dev/urandom
```

I wyjaśnienie użytych opcji:

smtpd_tls_auth_only	podczas autoryzacji TLS wymagany
smtp_use_tls smtpd_use_tls	włącz TLS
smtp_tls_note_starttls_offer	…
smtpd_tls_key_file smtpd_tls_cert_file smtpd_tls_CAfile	ścieżka do plików z kluczem i certyfikatem
smtpd_tls_loglevel	informacja diagnostyczna
smtpd_tls_received_header	czy w nagłówku są informacje dotyczące protokołu i szyfru
smtpd_tls_session_cache_timeout	czas wygasania sesji TLS
smtpd_tls_session_cache_database smtp_tls_session_cache_database	…
tls_random_source	źródło losowości

Tworzymy katalog /etc/postfix/ssl

Shell

```
# mkdir -p /etc/postfix/ssl
```

Przechodzimy do niego

```
Shell

# cd /etc/postfix/ssl
```

i robimy certyfikat

```
Shell

# openssl req -new -x509 -nodes -out smtpd.pem -keyout smtpd.pem -days 365
```

Przy tworzeniu certyfikatu wprowadzamy potrzebne dane:

```
Shell

Country Name (2 letter code) [AU]:PL
State or Province Name (full name) [Some-State]:POLAND
Locality Name (eg, city) []:Warszawa
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Moj certifikat
Organizational Unit Name (eg, section) []:POCZTA
Common Name (eg, YOUR name) []:moja-domena.pl
Email Address []:root@moja-domena.pl
```

UWAGA!!! Ważne żeby w polu **Common Name** podać własną domenę

Gdy juz wszystko mamy:

```
Shell

# /etc/init.d/postfix restart
```

4.5 Test TLS

Sprawdzamy czy działa

```
Shell

# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 moja-domena.pl ESMTP Postfix
EHLO moja-domena.pl
250-moja-domena.pl
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

Jeżeli widzimy linijkę która jest podkreślona znaczy że działa.

5. Zabezpieczenie serwera przed spamem i wirusami.

Oprócz kontroli poczty pod względem spamu na poziomie SMTPD, możliwa jest także ochrona z wykorzystaniem specjalnych programów, m.in. SpamAssassin. Aplikacja ta będzie współpracować z Postfix'em, Amavisd-new oraz Clam AV i poprawi bezpieczeństwo całego systemu.

5.1 Spamassassin

- *spamassassin* - Perl-based spam filter using text analysis
- *spamc* - Client for SpamAssassin spam filtering daemon
- *procmail* - Versatile e-mail processor

Shell

```
# aptitude install spamassassin spamc procmail
```

Aby Spamassassin startował razem z systemem przechodzimy do pliku:

Shell

```
# nano /etc/default/spamassassin
```

Znajdujemy linijkę ENABLED i edytujemy jak poniżej

File

```
ENABLED=1
```

Teraz możemy wystartować naszego spamassassina z defaultowym configiem lub stworzyć własny.

Shell

```
# nano /etc/spamassassin/local.cf
```

Teraz edytujemy dostępny config

File

```
# Wynik po ktorego otrzymaniu dostajemy wiadomosci oznaczone jako spam
required_score           5.0

# Temat wiadomosci oznaczonej jako spam
rewrite_header subject        **** UWAGA MOZLIWY SPAM ****

# Wiadomosc spam w zalaczniku na ustwiona nie (0=no, 1=yes, 2=safe)
report_safe             0

#Czy userzy moga tworzyc wlasne regulki w swoim katalogu domowym
allow_user_rules 0
```

```
# Bayes system
use_bayes               1

# Enable Bayes auto-learning
bayes_auto_learn        1

# Enable or disable network checks
skip_rbl_checks         0

# ustaw na 1 jezli masz zamiar odpalic razira
use_razor2              0

# Mail using locales used in these country codes will not be marked
# as being possibly spam in a foreign language.
# Mozesz dac np: pl en
ok_locales              all

# Tutaj wyedytowalem niektore regulki i zmienilem punktaje wystarczy je
wykasowac i bedzie defaultowo
score USER_IN_BLACKLIST 100000.000
score USER_IN_WHITELIST -100000.000
score UNWANTED_LANGUAGE_BODY 5
score HTML_IMAGE_ONLY_08 3
score FREE_TRIAL 5
score ALL_TRUSTED 0
score MICROSOFT_EXECUTABLE 4.5
score DATE_MISSING 1.5
score HTML_MESSAGE 0.5
score MIME_HTML_ONLY 0.5
score MIME_SUSPECT_NAME 0.5
score MISSING_MIMEOLE 1.5
score HTML_RELAYING_FRAME 2
score HTML_LINK_CLICK_HERE 3
score HTML_FONTCOLOR_BLUE 0.5
score HTML_FONTCOLOR_GREEN 0.5
score HTML_FONTCOLOR_RED 0.5
score HTML_20_30 1
score HTML_30_40 1
score HTML_40_50 1
score HTML_50_60 1
score HTML_60_70 1
score HTML_70_80 1
score HTML_80_90 1
score HTML_90_100 1
score CLICK_BELOW 3
score CLICK_BELOW_CAPS 3
score CLICK_TO_REMOVE_1 5
score CLICK_TO_REMOVE_2 5
score FOR_FREE 3
score NO_REAL_NAME 1.5
score PRIORITY_NO_NAME 1.5
score FORGED_YAHOO_RCVD 2
score FORGED_HOTMAIL_RCVD 2
score MISSING_OUTLOOK_NAME 0.5
score FORGED_OUTLOOK_TAGS 1.5
score LINES_OF_YELLING 0.5
score LINES_OF_YELLING_2 0.5
score LINES_OF_YELLING_3 0.5
score BIZ_TLD 1
score HEADER COUNT CTYPE 2.5
```

```
score MIME_HEADER_CTYPE_ONLY 2.5
score MORE_SEX 5

## white list ( tu wklejamy wazne adresy)
whitelist_from *@dug,net.pl

## black list (a tu nie chciane ktore spamassassin przepuszcza, a
ktorych niechcemy)
blacklist_from *@*.ru
blacklist_from *.@eu-vest.biz
```

startujemy spamassassina

Shell

```
# /etc/init.d/spamassassin start
```

W tym momencie musimy wysłać wiadomość testowa, sprawdzamy czy maile dochodzą jeśli tak możemy przejść do następnego kroku.

Tworzymy plik konfiguracyjny procmaila

Shell

```
# nano /etc/procmailrc
```

I dodajmy poniższe wartości:

File

```
# katalog glowny wiadomosci email
VERBOSE=off
MAILDIR=$HOME/Maildir
#LOGFILE=$MAILDIR/.procmail.log
DEFAULT=$MAILDIR/new

# pominiecie plikow konfiguracyjnych procmailrc w katalogach domowych
DROPPRIVS=yes

#regula skanowania poczty nie przekraczajacej 256kB

:0fw: spamassassin.lock
* < 256000
| /usr/bin/spamc

#przy 8 gwiazdkach spam trafia do kosza

:0
* ^X-Spam-Level: \*\*\*\*\*\*\*\*
/dev/null


# bugfix
:0
* ^^rom[ ] { LOG="*** Dropped F off From_ header! Fixing up. "
```

```
:0 fhw
| sed -e '1s/^/F/'
}
```

Jeśli nie korzystaliśmy z konfiguratora postfix nie wie co to procmail należy edytować plik.

Shell

```
# nano /etc/postfix/main.cf
```

i wklejamy taką linijkę

File

```
mailbox_command = procmail -a "$EXTENSION"
```

restartujemy postfixa

Shell

```
# /etc/init.d/postfix restart
```

Wysyłamy wiadomość testowa. W źródle maila który otrzymamy powinieneś zobaczyć takie linijki:

File

```
X-Spam-Checker-Version: SpamAssassin 3.1.7 (2006-10-05) on moja-domena.pl
X-Spam-Level:
X-Spam-Status: No, score=0.5 required=5.0 tests=DNS_FROM_RFC_ABUSE
   autolearn=no version=3.1.7
```

(Możemy wysłać wiadomość spam by sprawdzić czy na pewno spam zostanie ucięty)

5.2 Razor

- *razor* - spam-catcher using a collaborative filtering network

Shell

```
# aptitude install razor
```

tworzymy katalog domowy dla razora

Shell

```
# mkdir -p /etc/mail/spamassassin/.razor
```

Oraz wydajemy kolejno polecenia:

```
Shell

# razor-admin -home=/etc/mail/spamassassin/.razor -register
# razor-admin -home=/etc/mail/spamassassin/.razor -create
# razor-admin -home=/etc/mail/spamassassin/.razor -discover
```

następnie edytujemy config spamassassina'a

```
Shell

# nano /etc/spamassassin/local.cf
```

i aktywujemy w configu razor'a

```
File

use_razor2   1
razor_config /etc/mail/spamassassin/.razor/razor-agent.conf
```

edytujemy plik konfiguracyjny razora

```
Shell

# nano /etc/mail/spamassassin/.razor/razor-agent.conf
```

oraz dodajemy

```
File

razorhome = /etc/mail/spamassassin/.razor/
```

Możemy restartować spamassassina

```
Shell

# /etc/init.d/spamassassin restart
```

Teraz możemy wyedytować plik

```
Shell

# nano /etc/mail/spamassassin/v310.pre
```

Jeśli mamy hashe "#" na loadplugin razor2, to je usuwamy

File

```
loadplugin Mail::SpamAssassin::Plugin::Razor2
```

No i tak wzbogaciliśmy spamassassina o obsługę razor2.
Pamiętajmy o whitelistowaniu ważnych adresów dla firmy.
Zazwyczaj firmy z którymi korespondujemy nie zdają sobie sprawy, że są na czarnych listach serwerów.
Przez co wiadomości trafiają do /dev/null zamiast do naszych skrzynek.

5.3 Clamav i amavis

Spamassassin usunie spora część spamu, lecz sam spam nie jest groźny. Groźne są wirusy w załącznikach, które nieświadomi użytkownicy otwierają. By temu zapobiec możemy skanować pocztę w poszukiwaniu wirusów i ucinać maile z niechcianymi załącznikami za pomocą clamav i amavisa.

- *clamav-daemon* - antivirus scanner daemon
- *clamav* - antivirus scanner for Unix
- *amavisd-new* - Interface between MTA and virus scanner/content filters

Shell

```
# aptitude install clamav-daemon clamav amavisd-new
```

Edytujemy plik konfiguracyjny postfixa

Shell

```
# nano /etc/postfix/main.cf
```

i dodajemy na końcu linijkę

File

```
#Amavis
content_filter = smtp-amavis:[127.0.0.1]:10024
```

następnie edytujemy master.cf

Shell

```
nano /etc/postfix/master.cf
```

na samym końcu pliku wklejamy to:

File

```
##Amavis
smtp-amavis unix -      -       n        -      2       smtp
         -o smtp_data_done_timeout=1200s
         -o smtp_never_send_ehlo=yes
         -o disable_dns_lookups=yes
127.0.0.1:10025 inet n  -       n        -      -       smtpd
         -o content_filter=
         -o local_recipient_maps=
         -o realy_recipient_maps=
         -o smtpd_restriction_classes=
         -o smtpd_client_restrictions=
         -o smtpd_helo_restrictions=
         -o smtpd_sender_restrictions=
         -o smtpd_recipient_restrictions=permit_mynetworks,reject
         -o mynetworks=127.0.0.0/8
         -o strict_rfc821_envelopes=yes
         -o smtpd_error_sleep_time=0
         -o smtpd_soft_error_limit=1001
         -o smtpd_hard_error_limit=1000
```

Został nam clamav i amavis właściwie

Następnie edytujemy clamd.conf

Shell

```
# nano /etc/clamav/clamd.conf
```

Zmieniamy linie User na poniższą

File

```
User amavis
```

Edytujemy freshclam.conf

Shell

```
# nano /etc/clamav/freshclam.conf
```

Zmieniamy linie DatabaseOwner

File

```
DatabaseOwner amavis
```

Teraz zmieniamy prawa do odpowiednich katalogów

Shell

```
# chown -R amavis.amavis /var/run/clamav
# chown -R amavis.amavis /var/lib/clamav
# chown -R amavis.amavis /var/log/clamav
```

Musimy jeszcze poprawić logrotate (system archawizujacy logi w systemie).

Shell

```
# nano /etc/logrotate.d/clamav-daemon
```

Tak powinien wyglądać ten plik.

File

```
/var/log/clamav/clamav.log {
     rotate 12
     weekly
     compress
     delaycompress
     create 640 clamav adm
     postrotate
     /etc/init.d/clamav-daemon reload-log > /dev/null
     endscript
     }
```

I zmieniamy podkreśloną linię na poniższą

File

```
create 640 amavis amavis
```

Edytujemy pliki konfiguracyjne amavisa

Shell

```
# nano /etc/amavis/conf.d/15-content_filter_mode
```

I odznaczamy hashe "#" z linijek bypass_viruses

File

```
@bypass_virus_checks_maps = (
   \%bypass_virus_checks, \@bypass_virus_checks_acl, \$bypass_virus_checks_re);
```

Zakazane załączniki możemy znaleźć w pliku podanym poniżej. Edycja jest banalna wystarczy odchashować "#" linijke w której są rozszerzenia plików których nie chcemy w naszej poczcie, a zachaszowac te które chcemy ;)(ewentualnie usunąć jakieś rozszerzenie z linijki w której jest, lub dodać)

Shell

```
# nano /etc/amavis/conf.d/20-debian_defaults
```

Teraz resetujemy clamav'a i amavis'a

Shell

```
# /etc/init.d/clamav-daemon restart
# /etc/init.d/clamav-freshclam restart
# /etc/init.d/amavis restart
```

Jeżeli zobaczysz taki komunikat w logach wystarczy zwykły reset serwera

File

```
warning: connect to transpor smtp-amavis: No such file or directory
```

6. Uwagi

1. W Debian Etch plik konfiguracyjny amavisa znajduje sie w /etc/amavis/conf.d/ i jest podzielony na mniejsze pliki.
2. Jeśli chcesz zmienić serwer lustrzany lub ilość updatów clamav w ciągu dnia wystarczy wydać polecenie

```
Shell

# dpkg-reconfigure clamav-freshclam
```

3. Przypadkiem nie ustawiaj w amavisie odpowiedzi na mail z virusem.
4. Tak jak wyżej nie używaj softu do automatycznego odsyłania spamu.
5. Chcesz przetestować działanie swojego antyvirusa doinstaluj pakiet

```
Shell

# aptitude install clamav-testfiles
```

lub ściągnij sobie stad jeden z plików znajdujących się w tej paczce
http://www.netg.pl/~bialy/dug/amavis/test.exe
Dodaj do załącznika i wyślij. Root powinien dostać wiadomość o virusie.

6. Pamiętaj, ze antyvirus nie przeskanuje archiwum którego nie masz wiec jak chcesz skanować zip rar bzip musisz miec te archiwery.
7. Test spamassassina - jako treść wiadomości wpisz

```
File

XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X
```

Jeżeli niema cię na whiteliscie powinieneś zobaczyć info w logach o ilości pkt ile dostała ta wiadomość, u mnie jest "1000", oraz informacje o usunięciu tego maila.
7. Jeżeli masz zamiar zainstalować pyzora musisz wiedzieć, ze są problemy z instalacja. Należy ściągnąć odpowiednie łatki by zaczął funkcjonować.
8. Pełną listę komend spamassassina znajdziesz wydając polecenie

```
Shell

# perldoc Mail::SpamAssassin::Conf
```

7. Zarządzanie postfix'em

7.1 Squirrelmail'a czyli poczta przez WWW

Na koniec przydało by się zrobić użytkownika poczty dostęp przez WWW, a do tego najlepiej nadaje się squirrelmail, więc zainstalujmy go.

- *squirrelmail* – umożliwia dostęp do poczty przez www,
- *squirrelmail-locales* – paczka językowa do squirrelmail'a,

Shell
```
# aptitude install squirrelmail squirrelmail-locales
```

Po zainstalowaniu robimy symboliczne dowiązanie pliku apacze.conf

Shell
```
# ln -s /etc/squirrelmail/apache.conf /etc/apache2/conf.d/squirrelmail
```

I edytujemy plik:

Shell
```
# nano /etc/squirrelmail/apache.conf
```

Na samym początku tego pliku znajdujemy ta wartość:

File
```
Alias /squirrelmail /usr/share/squirrelmail
```

I zmieniamy jak poniżej

File
```
Alias /poczta /usr/share/squirrelmail
```

Zróbmy jeszcze dowiązanie do plików squirrelmail w katalogu WWW, żeby łatwiej było znaleźć:

Shell
```
# ln -s /usr/share/squirrelmail /var/www/poczta
```

Teraz reset apache:

Shell

```
# /etc/init.d/apache2 restart
```

I sprawdzamy czy wszystko działa wpisując poniższy adres w przegladarce (działa tylko na localhoscie):

http://localhost/squirrelmail/src/configtest.php
Powinieneś zobaczyć kilka komunikatów takich jak:

File

```
Checking PHP configuration...
    PHP version 5.2.0-8+etch4 OK.
    PHP extensions OK.
Checking paths...
    Data dir OK.
    Attachment dir OK.
    Plugins are not enabled in config.
    Themes OK.
    Default language OK.
    Base URL detected as: http://localhost/poczta/src (location base autodetected)
Checking outgoing mail service....
    SMTP server OK (220 moja-domena.pl ESMTP Postfix)
Checking IMAP service....
    IMAP server ready (* OK POP ready)
    Capabilities: * CAPABILITY IMAP4rev1 SASL-IR SORT THREAD=REFERENCES MULTIAPPEND
UNSELECT LITERAL+ IDLE CHILDREN NAMESPACE LOGIN-REFERRALS STARTTLS AUTH=PLAIN
Checking internationalization (i18n) settings...
     gettext - Gettext functions are available. You must have appropriate system
locales compiled.
     mbstring - Mbstring functions are available.
     recode - Recode functions are unavailable.
     iconv - Iconv functions are available.
     timezone - Webmail users can change their time zone settings.
Checking database functions...
    not using database functionality.

Congratulations, your SquirrelMail setup looks fine to me!
```

Jeżeli wszystko jest OK. znaczy że możesz się już załogować do poczty, a jeżeli masz błędy musisz edytować plik konfiguracyjny squirrelmail'a komendą:

Shell

```
# /usr/sbin/squirrelmail-configure
```

Po uruchomieniu tego skryptu zobaczysz takie "okno" i edytujemy co trzeba

Shell

```
SquirrelMail Configuration : Read: config.php (1.4.0)
---------------------------------------------------------
Main Menu --
1.  Organization Preferences
2.  Server Settings
3.  Folder Defaults
4.  General Options
5.  Themes
6.  Address Books
7.  Message of the Day (MOTD)
8.  Plugins
9.  Database
10. Languages

D.  Set pre-defined settings for specific IMAP servers

C   Turn color on
S   Save data
Q   Quit

Command >>
```

8. Monitoring

8.1 Mailgraph czyli wykresy

Mailgraph tworzy dzienne, tygodniowe, miesięczne i roczne wykresy z wysłanych odebranych, bounced i odrzuconych wiadomości także spamu i wirusów jeżeli SpamAssassin i ClamAV jest zintegrowany z Postfixem.

- *rrdtool* - Time-series data storage and display system (programs),
- *mailgraph* - Mail statistics RRDtool frontend for Postfix

Shell

```
# aptitude install rrdtool mailgraph
```

Podczas instalacji będziesz pytany o kilka rzecz zaznacz jak poniżej: (Jeżeli masz zintegrowanego z Postfix'em amavis'a do filtrowania spamu i wirusów wtedy w opcji *Count incoming mail as outgoing mail?* zaznacz *No* żeby uniknąć podwójnego liczenia maili (ponieważ Postfixa dostarcza maile do amavis'a który po dokonaniu skanowania odsyła je z powrotem do Postfix'a). Jeżeli nie używasz żadnego skanera wtedy możesz zaznaczyć *Yes*)
Jeżeli instalator nie zapytał cię o te opcje podaj wydaj polecenie *dpkg-reconfigure mailgraph*

File

```
Should Mailgraph start on boot?                 <-- Yes
Which logfile should be used by mailgraph?      <-- /var/log/mail.log
Count incoming mail as outgoing mail?           <-- Yes
```

Utwórz katalog

Shell

```
# mkdir -p /var/www/cgi-bin/
```

Teraz kopiujemy skrypt odpowiedzialny za generowanie wykresu do katalogu stworzonego przez nas:

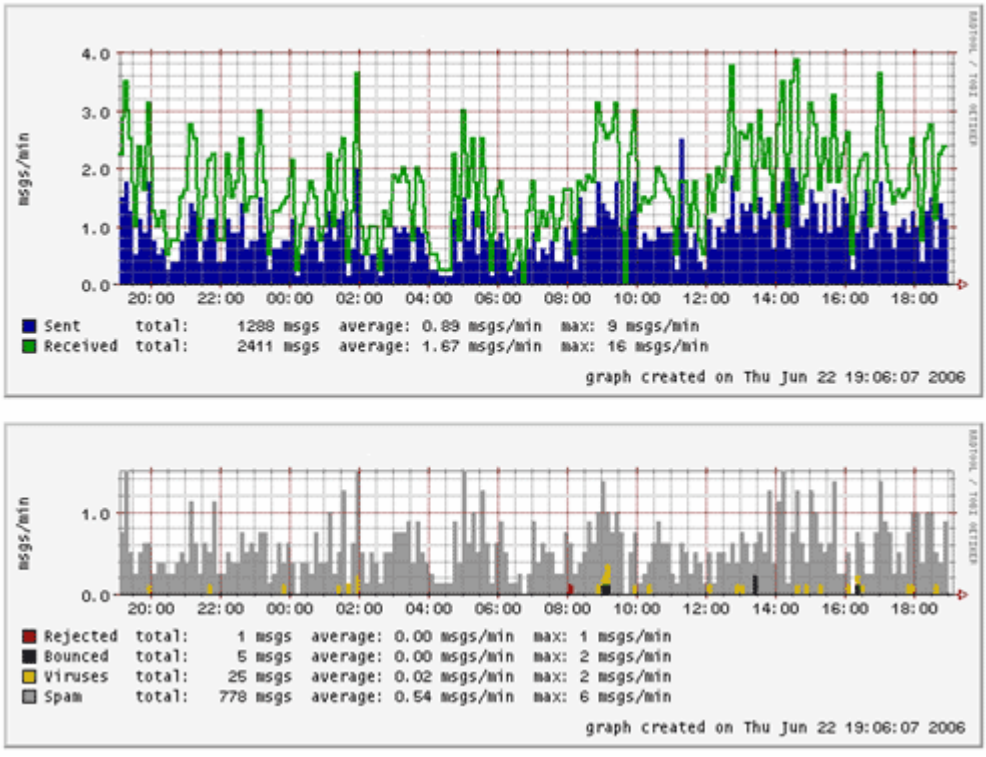
Shell

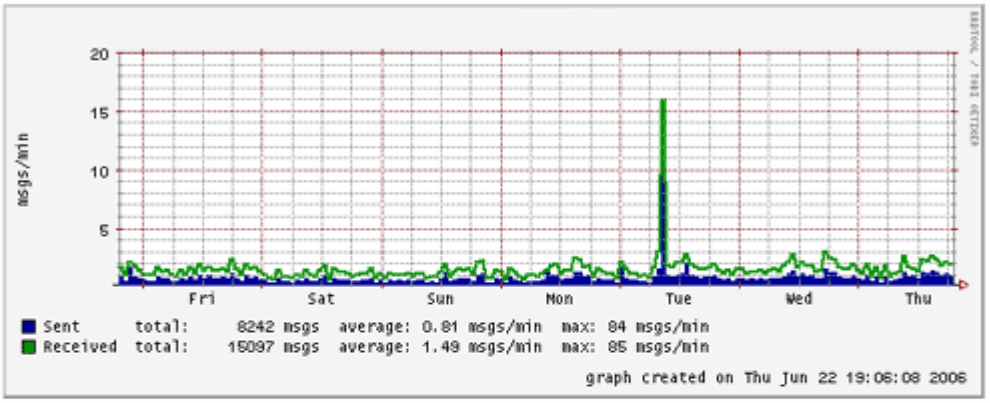
```
# cp -p /usr/lib/cgi-bin/mailgraph.cgi /var/www/cgi-bin/
```

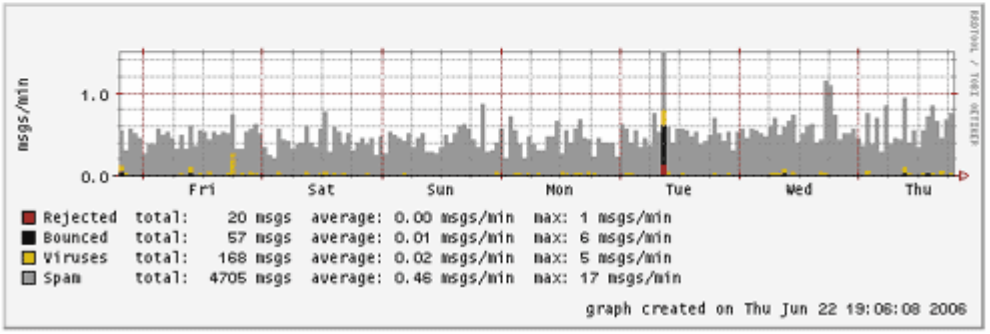
Żeby zobaczyć nasze wykresy wpisujemy w przeglądarce:

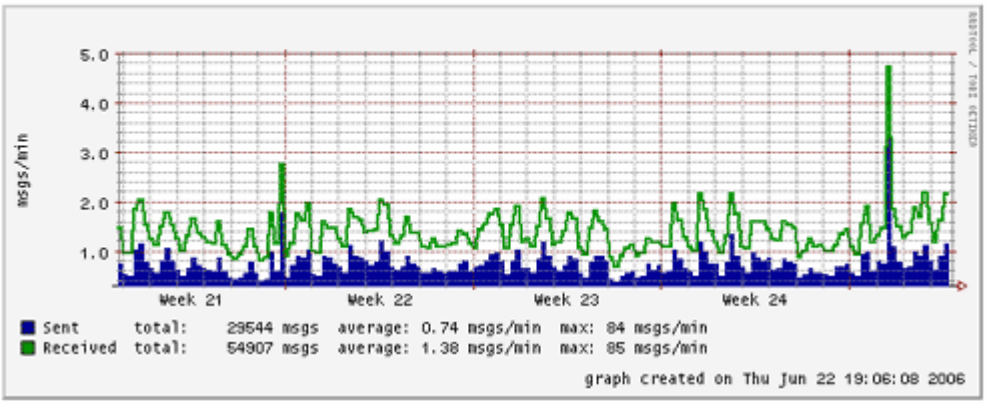
http://localhost/cgi-bin/mailgraph.cgi

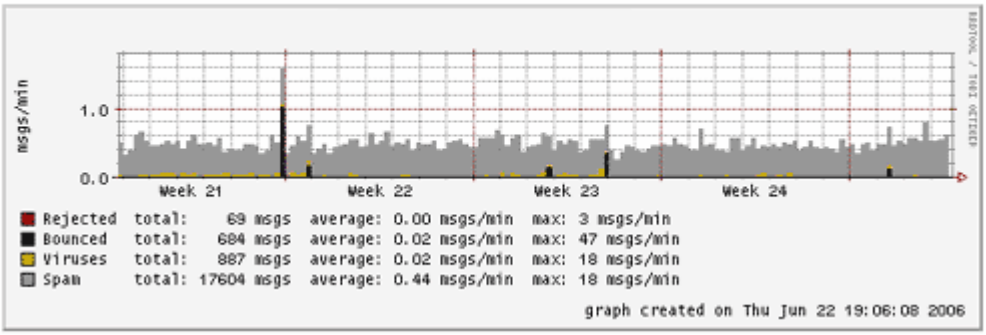
I powinniśmy ujrzeć wykresy:

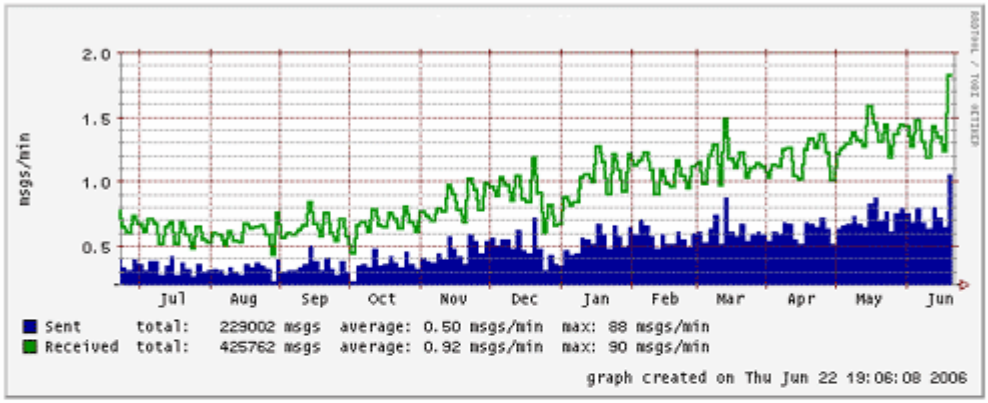


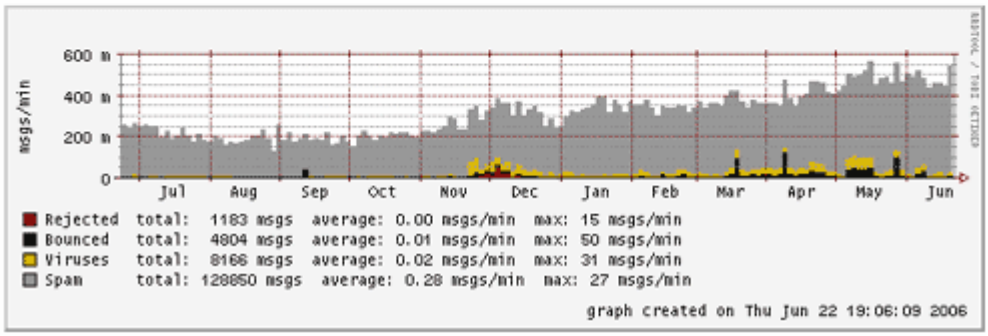

msgs/min
20
15
10
5
Fri
Sat
Sun
Mon
Tue
Wed
Thu
Sent total: 8242 msgs average: 0.81 msgs/min max: 84 msgs/min
Received total: 15097 msgs average: 1.49 msgs/min max: 85 msgs/min
graph created on Thu Jun 22 19:06:08 2006


msgs/min
1.0
0.0
Fri
Sat
Sun
Mon
Tue
Wed
Thu
Rejected total: 20 msgs average: 0.00 msgs/min max: 1 msgs/min
Bounced total: 57 msgs average: 0.01 msgs/min max: 6 msgs/min
Viruses total: 168 msgs average: 0.02 msgs/min max: 5 msgs/min
Spam total: 4705 msgs average: 0.46 msgs/min max: 17 msgs/min
graph created on Thu Jun 22 19:06:08 2006


msgs/min
5.0
4.0
3.0
2.0
1.0
Week 21
Week 22
Week 23
Week 24
Sent total: 29544 msgs average: 0.74 msgs/min max: 84 msgs/min
Received total: 54907 msgs average: 1.38 msgs/min max: 85 msgs/min
graph created on Thu Jun 22 19:06:08 2006


msgs/min
1.0
0.0
Week 21
Week 22
Week 23
Week 24
Rejected total: 69 msgs average: 0.00 msgs/min max: 3 msgs/min
Bounced total: 684 msgs average: 0.02 msgs/min max: 47 msgs/min
Viruses total: 887 msgs average: 0.02 msgs/min max: 18 msgs/min
Spam total: 17604 msgs average: 0.44 msgs/min max: 18 msgs/min
graph created on Thu Jun 22 19:06:08 2006





8.2 Pflogsumm czyli raport

Pflogsumm jest skryptem napisanym w perlu do generowania statystyk z logów maila. Mamy takie statystki jak: z którego konta zostało wysłano najwięcej maili, kto do nas wysłał najwięcej maili, w których godzinach jest wysyłane najwięcej mail itd. …

Przedstawię tu jak zainstalować i skonfigurować ten skrypt żeby otrzymywać raz w tygodniu raport z wysłanych i otrzymanych mail.

Zainstalujemy ten skrypcik:

- *pflogsumm* - Postfix log entry summarizer

```
Shell
# aptitude install pflogsumm
```

Zanim zajmiemy się konfiguracją skryptu najpierw musimy skonfigurować, aby logi mail były pakowane raz dziennie w tym celu tworzymy plik:

```
Shell
# nano /etc/logrotate.d/mail
```

I dodajemy poniższa zawartość

File

```
/var/log/mail.log {
    missingok
    daily
    rotate 9
    create
    compress
    start 0
}
```

Tworzymy plik:

Shell

```
# nano /root/bin/postfix_report.sh
```

Dodajemy poniższa zawartość i zmieniamy podkreślone wartości:

File

```
#!/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
gunzip /var/log/mail.log.0.gz

pflogsumm /var/log/mail.log.0 | formail -c -I"Subject: Mail Statistics" -I"From:
pflogsumm@moja-domena.pl" -I"To: postmaster@moja-domena.pl" -I"Received: from
www.moja-domena.com" | sendmail postmaster@moja-domena.pl

gzip /var/log/mail.log.0
exit 0
```

Nadajmy prawa do wykonywania dla tego pliku:

Shell

```
# chmod 750 /root/bin/postfix_report.sh
```

Otwórzmy plik cron'a:

Shell

```
# nano /etc/crontab
```

Dodajmy go do crona aby wykonywał się codziennie o 7:00 i wysyłał nam raport na podany przez nasz adres w skrypcie.

File

```
0 7 * * * /root/bin/postfix_report.sh &> /dev/null
```

9. Linki

Korzystałem z tego opisu (HOW TO Postfix + sasl + tls on Debian Etch by BialyS)

- http://forum.dug.net.pl/viewtopic.php?t=6331

Bardzo dobra strona o zabezpieczeniu postfixa.

- http://lemat.priv.pl/index.php?m=page&pg_id=90

Dużo ciekawych artykułów po angielsku

- http://www.howtoforge.com/