

UNIwersytet w Białymstoku

Wydział Matematyczno-Fizyczny

Instytut Matematyki

Paweł Kiersnowski

SYSTEM ZARZĄDZANIA
POCZTOWYMI HOSTAMI
WIRTUALNYMI
NA SERWERZE ISP

*Praca dyplomowa napisana
pod kierunkiem
dr. hab. Krzysztofa Prażmowskiego*

Białystok 2003

Składam serdeczne podziękowania
mgr Mariuszowi Żynelowi za pomoc
podczas przygotowywania niniejszej pracy.

Paweł Kiersnowski

Spis treści

Wstęp	1
1 Zasady działania serwera SMTP na przykładzie Sendmail	2
1.1 Poczta elektroniczna	2
1.2 Protokół SMTP	4
1.3 Budowa przesyłki e-mail	6
1.3.1 Zapis adresu	6
1.3.2 Anatomia nagłówka pocztowego	7
1.3.3 Pola świadczące o drodze przesyłania listu	8
1.3.4 Pola nadawcy	9
1.3.5 Pola odbiorcy	9
1.3.6 Pola referencyjne	10
1.3.7 Pozostałe pola	10
1.4 Rekordy MX	10
2 Hosty wirtualne	13
2.1 Konta wirtualne	13
2.2 Konfiguracja Sendmaila	15
2.3 Odbieranie i wysyłanie poczty	16
3 Programy administracyjne	18
3.1 Konfiguracja oprogramowania	18
3.2 Obsługa wirtualnych domen	19
3.2.1 Dodanie nowej domeny	19
3.2.2 Usunięcie domeny	20
3.3 Obsługa wirtualnych kont	20
3.3.1 Założenie konta wirtualnego użytkownika	20
3.3.2 Usunięcie konta wirtualnego użytkownika	20
3.3.3 Utworzenie wirtualnego aliasu	21
3.3.4 Usunięcie wirtualnego aliasu	21
A Skrypty	22
Spis literatury	31

Wstęp

Poczta elektroniczna (z języka angielskiego: *e-mail* lub *electronic mail*) jest jedną z wielu usług w sieci Internet, służącą do przekazywania osobistych wiadomości pomiędzy użytkownikami w sieci. Wiele milionów osób korzystających z sieci Internet, posiada swoje własne adresy pocztowe. Rosnąca ilość bramek pocztowych powoduje, iż każdego dnia wzrasta liczba osób mogących korzystać z poczty elektronicznej w Internecie. *E-mail* ma pewną przewagę nad telefonem: można przygotować i wysłać list kiedy ma się na to czas i ochotę. Adresat listu może na niego odpowiedzieć w dogodnym dla niego momencie. Ponadto, dzwoniąc za granicę, dość szybko "nabijamy licznik" naszego rachunku, zaś dzięki poczcie elektronicznej można komunikować się z ludźmi mieszkającymi na całym świecie płacąc jedynie za lokalne połączenie telefoniczne, bądź stałą zryczałtowaną opłatę za korzystanie z internetu.

W rozdziale pierwszym przedstawiony jest przykład przedstawiający działanie poczty elektronicznej od strony praktycznej. Znajduje się tam również dokładny opis przesyłki e-mail. Oprócz tego pierwszy rozdział zawiera opis protokołu SMTP i jego konfigurację.

W rozdziale drugim umieszczono zagadnienia związane z hostami wirtualnymi. Znajduje się tam opis konta wirtualnego oraz szczegóły konfiguracji hostów wirtualnych przy zastosowaniu serwera SMTP Sendmail. Przedstawiona jest tam również baza danych MySQL zawierająca informacje o kontaktach wirtualnych oraz opis interfejsu WWW do tej bazy.

W ostatnim trzecim rozdziale zebrano opisy opracowanych programów, które usprawniają pracę Administratora. Programy te ułatwiają zakładanie i usuwanie poszczególnych wirtualnych domen, kont oraz wirtualnych użytkowników. Nie tylko opracowane programy, ale również opisy protokołu SMTP oraz konfiguracji hostów wirtualnych w Sendmail, powinny być pomocne w pracy Administratora serwera pocztowego.

Rozdział 1

Zasady działania serwera SMTP na przykładzie Sendmail

1.1 Poczta elektroniczna

Poczta elektroniczna z założenia jest bardzo podobna do zwykłej poczty, którą wysyła się do znajomych czy przyjaciół. Poczta elektroniczna, którą wysyła się do znajomych, przesyłana jest na ich prywatny, unikalny adres. Podobnie, gdy oni odpisują, poczta kierowana jest na prywatny adres oryginalnego nadawcy. Podobnie jak "na poczcie" można subskrybować elektroniczne wydania czasopism i gazet. Poczta elektroniczna ma dwie cechy które dają jej zasadniczą przewagę nad zwykłą pocztą. Najważniejszą z nich jest prędkość. W przeciwieństwie do zwykłej poczty, gdzie listy "idą" kilka dni bądź tygodni, przesyłki elektroniczne dostarczane są do adresata w ciągu kilku, kilkunastu minut (w zależności od stanu technicznego sieci). Drugą cechą wyróżniającą pocztę elektroniczną jest możliwość wykorzystywania tejże poczty do przeglądania baz danych, katalogów, przesyłania plików.

Załóżmy, że w pewnym mieście mamy dwa urzędy pocztowe: *onet.pl* i *wp.pl*. W którymś z nich każdy z mieszkańców ma swoją osobistą skrytkę pocztową (zamykaną na kluczyk). Przekazanie wiadomości listownej może przebiegać na dwa sposoby.

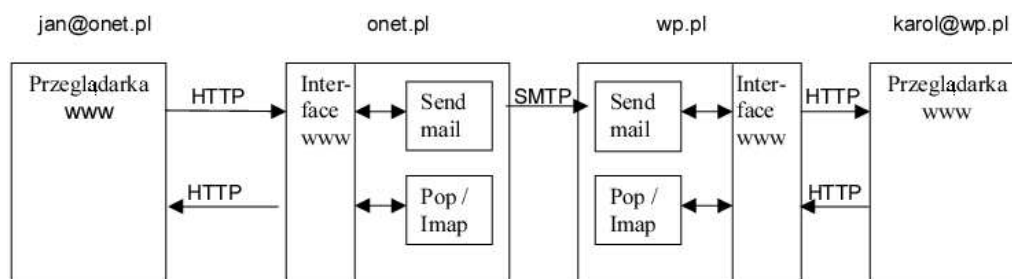
1. Jan, który korzysta ze skrytki *jan* na poczcie *onet.pl* postanawia wysłać wiadomość do Karola, który jest użytkownikiem skrytki pocztowej *karol* znajdującej się na poczcie *wp.pl*. W tym celu Jan udaje się do swojego urzędu pocztowego, pisze list, wkłada go do koperty, umieszcza na niej adres *karol@wp.pl* i wrzuca do skrzynki pocztowej. Co jakiś czas urzędnik poczty opróżnia skrynkę i sortuje listy. Te, które w adresie mają *onet.pl* umieszcza w odpowiednich skrytkach pocztowych. Pozostałe zaś, w tym list do Karola, przesyła do urzędu pocztowego *wp.pl*. List od Jana jest następnie przez obsługę tej poczty wrzucany do skrytki Karola, skąd jest zabierany przez jej właściciela.

2. Jan pisze list do Karola w swoim domu. Następnie wkłada go do zaadresowanej koperty i poleca gońcowi wrzucenie listu do skrzynki na poczcie. List po dotarciu do urzędu pocztowego *wp.pl* jest umieszczany w odpowiedniej skrytce skąd (upoważniony) gońiec, bądź listonosz dostarcza list do domu Karola.

Częścią wspólną tych dwóch różnych scenariuszy jest zastosowanie protokołu SMTP podczas wymiany poczty.

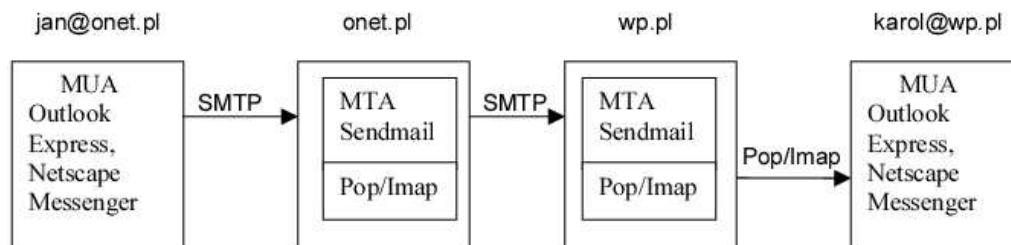
Pierwszy sposób komunikacji odpowiada sytuacji, kiedy Jan jako użytkownik serwera pocztowego *onet.pl* rejestruje się na stronie *onet.pl* (loguje się jako *jan*) i przy pomocy interfejsu „wrzuca list do skrzynki”, czyli przekazuje go odpowiedniemu programowi do obsługi poczty (najczęściej tym programem jest *Sendmail*). Taki program następnie analizuje adres i decyduje, czy list ma zostać wrzucony do lokalnej skrytki pocztowej, czy też musi zostać przekazany innemu serwerowi pocztowemu. Karol jako użytkownik serwera pocztowego *wp.pl*, aby sprawdzić czy jest dla niego poczta, udaje się do swojego „urzędu pocztowego” czyli rejestruje się na stronie *wp.pl* (loguje się jako *karol*) i sprawdza zawartość skrytki.

W powyższym przykładzie list zostaje wysłany na adres *karol@wp.pl*, co oznacza, że serwer pocztowy odpowiedzialny za domenę *onet.pl* nawiązuje połączenie TCP/IP z serwerem obsługującym domenę *wp.pl* i zgodnie z protokołem SMTP przekazuje list. Po odebraniu listu serwer domeny *wp.pl* umieszcza list w skrytce pocztowej *karol*. Poniższy schemat przedstawia tę sytuację:



Drugi sposób komunikacji odpowiada sytuacji, gdy Jan wykorzystuje swój komputer do uruchomienia programu pocztowego (np. Outlook Express, Netscape Messenger, Pegasus, itp.), przy pomocy którego redaguje list, podaje adres pod który list ma zostać doręczony i przesyła go do serwera pocztowego, na którym ma skrytkę pocztową. Do przekazania listu pomiędzy komputerem użytkownika i serwerem pocztowym wykorzystywany jest protokół SMTP tak jak przy wymianie poczty pomiędzy serwerami *onet.pl* i *wp.pl*. Karol, aby sprawdzić czy jest dla niego poczta, na swoim komputerze uruchamia program pocztowy taki jak np. Outlook Express czy Netscape Messenger. Przy pomocy tego programu sprawdza, czy w jego skrytce są nowe listy i pobiera je sobie do domu, czyli umieszcza na swoim komputerze. Operacja sprawdzania

i pobierania poczty jest wykonywana w oparciu o protokół POP (ang. *Post Office Protocol*) [5] lub IMAP (ang. *Internet Mail Access Protocol*) [4]. W tej sytuacji działanie poczty elektronicznej można przedstawić przy pomocy następującego schematu:



1.2 Protokół SMTP

SMTP (ang. *Simple Mail Transfer Protocol*) [6] czyli Prosty Protokół Przekazywania Poczty to protokół komunikacyjny opisujący sposób przekazywania poczty elektronicznej w Internecie. Działa on w technologii *klient-serwer*. Maszyna nadająca wiadomość jest zwana *klientem*, natomiast maszyna odbierająca nią jest zwana *serwerem SMTP*. W ten sposób nazywa się również oprogramowanie biorące udział w procesie przekazywania poczty. W stosunku do klienta, który sam nie jest serwerem stosuje się termin MUA (ang. *Mail User Agent*), zaś do serwera MTA (ang. *Mail Transport Agent*).

Sendmail [2] był jednym z pierwszych (jeśli nie pierwszym) z programów do przesyłania poczty, w którym zastosowano SMTP. Aktualnie istnieje wiele programów implementujących SMTP jako klient (nadawca) lub serwer (odbiorca wiadomości). Przykładem MTA oprócz Sendmaila jest: Qmail, Procmail, Exim, Zmailer, natomiast znanymi MUA są: MS Outlook, Netscape Messenger, KMail, Pine, The Bat.

SMTP zaczęło być szeroko używane we wczesnych latach osiemdziesiątych. W tamtym okresie było to uzupełnienie protokołu UUCP (ang. *Unix to Unix Copy Protocol*), który lepiej sprawdzał się przy przekazywaniu poczty między maszynami posiadającymi jedynie okresowe połączenie. SMTP natomiast lepiej działa, gdy zarówno maszyna nadająca jak odbierająca są na stałe przyłączone do sieci.

Protokół SMTP nie radził sobie dobrze z plikami binarnymi, ponieważ jest stworzony w oparciu o czysty 7-bitowy tekst ASCII (a jak wiadomo przy pomocy poczty elektronicznej można przysyłać nie tylko informacje tekstowe, ale wszelkie dane binarne). Trzeba to jednak robić w postaci tzw. załączników, czyli danych binarnych, które na potrzeby poczty elektronicznej są przekształcane do postaci tekstowej (ASCII). Służy do tego metoda zwana base64, która zamienia grupy 3 kolejnych bajtów w 4 znaki ASCII. Wszystkie nowoczesne

programy pocztowe pozwalają na odczytywanie załączników do listów i równie łatwe dołączanie dowolnych plików z danymi do listów. Problem może powstać jedynie wówczas, gdy dwa programy pocztowe (nadawcy i adresata) stosują odmienne zasady kodowania danych. Obecnie najpopularniejszym standardem rozwiązującym ten problem i umożliwiającym przesyłanie dowolnych danych, w tym listów zawierających polskie litery (a funkcjonuje około 20 sposobów kodowania polskich znaków!) jest standard MIME (ang. *Multipurpose Internet Mail Extension*) - uniwersalne rozszerzenie poczty internetowej. Jeśli przesyłany plik jest duży, to warto przed przesłaniem poddać go kompresji. W dzisiejszych czasach większość serwerów SMTP obsługuje rozszerzenie 8BITMIME pozwalające przysyłać pliki binarne równie łatwo jak tekst.

Jednym z ograniczeń pierwotnego SMTP jest brak mechanizmu weryfikacji nadawcy. Żeby temu zaradzić stworzono rozszerzenie SMTP-AUTH.

Łatwo przetestować serwer SMTP przy użyciu programu `telnet`. Standardowym portem, na którym nasłuchuje serwer SMTP to port 25. Przekazywanie wiadomości pomiędzy klientem a serwerem następuje w drodze dialogu. To jakich słów (poleceń) w tym dialogu możemy użyć określone zostało w specyfikacji protokołu SMTP [6]. Przykładowy dialog może wyglądać następująco (znakiem `>` oznaczono wprowadzane polecenia SMTP):

```
> telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 math.uwb.edu.pl ESMTP Sendmail 8.11.4/8.11.4; Fri, 16 May 2003
14:32:16 +0200 (CEST)
> helo uwb.edu.pl
250 math.uwb.edu.pl Hello localhost [127.0.0.1], pleased to meet
you
> mail from: kiersno@theta.uwb.edu.pl
250 2.1.0 kiersno@theta.uwb.edu.pl... Sender ok
> rcpt to: mariusz@math.uwb.edu.pl
250 2.1.5 mariusz@math.uwb.edu.pl... Recipient ok
> data
354 Enter mail, end with "." on a line by itself
>
> Test poczty
>
> .
250 2.0.0 h4GCW1401685 Message accepted for delivery
> quit
221 2.0.0 math.uwb.edu.pl closing connection
Connection closed by foreign host.
```

W efekcie na konto `mariusz@math.uwb.edu.pl` wpłynął e-mail o następującej postaci:

Received: from uwb.edu.pl (localhost [127.0.0.1])
by math.uwb.edu.pl (8.11.4/8.11.4) with SMTP id h4GCW1401685
for mariusz@math.uwb.edu.pl; Fri, 16 May 2003 14:32:59 +0200 (CEST)
Date: Fri, 16 May 2003 14:32:59 +0200 (CEST)
From: kiersno@theta.uwb.edu.pl
Message-Id: <200305161232.h4GCW1401685@math.uwb.edu.pl>
Content-Length: 13
Test poczty

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.

Program *Sendmail* jest najbardziej rozpowszechnionym programem MTA. Do podstawowych jego funkcji należy:

- akceptacja poczty użytkownika,
- interpretacja adresu pocztowego,
- przepisanie adresu na postać właściwą dla danego programu przenoszącego pocztę i przekazanie mu poczty,
- przesyłanie i odbieranie poczty poprzez Internet,
- możliwość tworzenia list adresowych, stosowania przydomków dla indywidualnych użytkowników,
- automatyczne przekazywanie poczty do innych serwerów (poprzez zapisy w pliku *forward*) i filtrowanie poczty (dzięki pośrednictwu programu *procmail*).

1.3 Budowa przesyłki e-mail

1.3.1 Zapis adresu

Wysyłając list na zewnątrz systemu należy podać w polu To: adres internetowy adresata. Adres taki wygląda następująco:

kiersno@theta.uwb.edu.pl

Ciąg znaków przed znakiem "at" @ jest nazwą użytkownika, zaś napis za tym znakiem jest nazwą domeny lub komputera, gdzie dany użytkownik ma swoje konto. Z adresu tego można wyczytać, że mam konto na komputerze należącym do sieci o domenie *uwb.edu.pl* (Uniwersytet w Białymstoku). Duże organizacje, uczelnie czy też firmy, mają swoje sieci podzielone na podsieci, w tych zaś wyodrębnione są poszczególne komputery. W moim przypadku konto mam na komputerze o nazwie *theta*.

1.3.2 Anatomia nagłówka pocztowego

Oprócz samej wiadomości, do każdego listu dodawane są jeszcze nagłówki. Odpowiednikiem nagłówka w zwykłej poczcie może być "stempel". Nagłówki tworzą kilka bądź kilkanaście pierwszych linii listu. Każdy system, przez który przesyłana jest wiadomość dalej, dodaje przynajmniej jeden nagłówek ("stempluje list"). Z tychże "stempli" można wyczytać bardzo wiele informacji: przez kogo i kiedy został wysłany, kiedy został odebrany i gdzie przekazany list.

Każda przesyłka składa się z pól nagłówka (header) i treści (body). Treść, z punktu widzenia poczty elektronicznej, jest zbiorem linii zawierających znaki ASCII, zaś od nagłówka oddziela ją pusta linia zakończona CRLF. Każde pole nagłówka może być traktowane jako pojedyncza, samodzielna linia, złożona ze znaków ASCII zawierająca nazwę pola i zawartość pola. Zawartość pola może być podzielona na kilka linii, np.:

```
To: "Paweł Kiersnowski" kiersno@theta.uwb.edu.pl
```

może być przedstawione jako:

```
To: "Paweł Kiersnowski"  
kiersno@theta.uwb.edu.pl
```

lub jako:

```
To: "Paweł  
Kiersnowski" kiersno@theta.uwb.edu.pl
```

Przykład oryginalnego nagłówka przedstawiono poniżej:

```
Received: from mailgate.uwb.edu.pl (mailgate [212.33.71.77])  
by math.uwb.edu.pl (8.11.4/8.11.4) with ESMT  
id h4FA7o405154  
for <mariusz@math.uwb.edu.pl>;  
Thu, 15 May 2003 12:07:50 +0200 (CEST)  
Received: from smtp.wp.pl (smtp.wp.pl [212.77.101.161])  
by mailgate.uwb.edu.pl (8.11.6p2/8.11.6) with ESMT  
id h4FA7kT08886  
for <mariusz@math.uwb.edu.pl>;  
Thu, 15 May 2003 12:07:46 +0200  
Received: (WP-SMTPD 15041 invoked from network); 15 May 2003  
10:07:34-0000  
Received: from ofree.wp-sa.pl (HELO localhost) ([212.77.101.203])  
(envelope-sender <kiersno.p@wp.pl>)  
by smtp4.free.wp-sa.pl (wp-smtpd) with SMTP  
for <mariusz@math.uwb.edu.pl>; 15 May 2003 10:07:34 -0000  
Date: Thu, 15 May 2003 12:07:34 +0200  
From: =?ISO-8859-2?Q?Pawe=B3_Kiersnowski?= <kiersno.p@wp.pl>  
To: "Mariusz Zynel" <mariusz@math.uwb.edu.pl>
```

Subject: Przesyłam prace
Message-ID: <3ec366e6325b1@wp.pl>
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="part3ec366e637360"
X-Mailer: Interfejs WWW poczty Wirtualnej Polski
Organization: Poczta Wirtualnej Polski S.A. <http://www.wp.pl/>
X-IP: 217.97.90.77, 192.168.4.30
X-AntiVirus: skaner antywirusowy poczty Wirtualnej Polski S. A.
X-WP-ChangeAV: 0
X-RAVMilter-Version: 8.4.2(snapshot 20021217) (mailgate)
Content-Length: 45029

This is a multi-part message in MIME format.

--part3ec366e637360
Content-Type: text/plain; charset=iso-8859-2
Content-Transfer-Encoding: 8bit

Tutaj znajduje się treść przesyłki...

1.3.3 Pola świadczące o drodze przesyłania listu

Return-Path — Adres powrotny; Pole to jest dodawane przez ostatni system biorący udział w przesyłaniu informacji, system, który dostarcza przesyłkę do odbiorcy. Pole to służy do określenia informacji o adresie i drodze do nadawcy. Zawartość pola **Reply-To** jest dodawana przez nadawcę i służy do wskazania adresu, na który mają być wysyłane odpowiedzi, podczas gdy **Return-Path** służy identyfikacji drogi powrotnej do nadawcy.

Received — Pole to dodawane jest przez każdy host biorący udział w przesyłaniu wiadomości. Informacje z tego pola mogą być bardzo pomocne w lokalizowaniu i rozwiązywaniu problemów związanych z transportem przesyłki. W polu tym mogą być podane nazwy hostów: wysyłającego i odbierającego oraz czas otrzymania przesyłki. Dodatkowo parametr **via** może być użyty do wskazania konkretnego mechanizmu fizycznego użytego do przesłania wiadomości, zaś parametr **with** wskazuje na protokół użyty do przesłania przesyłki, np.: SMTP, X.25.

Ogólna budowa tego nagłówka jest następująca :

```
Received: from    <host wysyłający>
              by    <host odbierający>
[via            <droga przesyłki>]
              with <protokół>
              id    <numer przesyłki>
```

```
for      <adres pocztowy adresata>;  
        <data i czas>
```

W przykładzie podanym w 1.3.2, pierwsze pole należy czytać następująco:

przesyłka o numerze [id] h4FA7o405154 przesyłana była we wtorek, 15 maja 2003 roku do adresata posiadającego adres e-mail <mariusz@math.uwb.edu.pl> [for] przy użyciu [with] protokołu ESMTP, przy czym komputerem wysyłającym był komputer o nazwie mailgate.uwb.edu.pl [from] zaś komputerem odbierającym jest math.uwb.edu.pl [by].

Śledząc kolejne pola **Received** jesteśmy w stanie określić, że oryginalnym serwerem SMTP, z którego pochodzi przesyłka to *ofree.wp-sa.pl*. Następnym hostem do którego tę przesyłkę przekazano to *smtp4.free.wp-sa.pl*, potem do *smtp.wp.pl* i dalej do *mailgate.uwb.edu.pl*. Hostem przeznaczenia przesyłki jest *math.uwb.edu.pl*.

1.3.4 Pola nadawcy

From — Identyfikator autora. Pole to informuje o autorze listu, zawartość pola tworzona jest w czasie tworzenia przesyłki - stanowi ją adres sieciowy twórcy listu (może być to adres użytkownika, systemu lub procesu tworzącego list). Jeżeli w czasie tworzenia listu pole to nie jest dodawane, w nagłówku powinno znaleźć się pole **Sender**

Sender — Identyfikator nadawcy listu; Pole to informuje o agencji (osobie, systemie lub procesie) wysyłającej list. Pole to zostało wprowadzone dla odróżnienia nadawcy od autora listu. Jeżeli pola **From** i **Sender** są takie same, pole **Sender** nie musi występować w nagłówku.

Reply-To — Pole wskazuje do kogo powinny być wysyłane odpowiedzi na dany list. W przeciwieństwie do dwu powyższych pól: **From** i **Sender** pole **Reply-To** może być ustawione przez nadawcę listu. Najczęściej pole to zawiera adres sieciowy nadawcy listu, jednak może się zdarzyć, że nadawca chce otrzymać odpowiedź na inny adres sieciowy, lub też aby z odpowiedzią na list zapoznała się większa grupa osób (cecha ta może być przydatna w dyskusji - konferencji prowadzonej za pomocą poczty elektronicznej).

1.3.5 Pola odbiorcy

To — Pole to zawiera informacje o głównych adresatach listu.

Cc — Pole to zawiera informacje o adresatach listu, do których dany list został przesłany jako kopia. W korespondencji papierowej odpowiednikiem tego pola jest rozdzielnik „do wiadomości”.

Bcc — Pole to zawiera informacje o dodatkowych adresatach listu. Zawartość tego pola nie jest wyświetlana w listach wysłanych do głównych adresatów **To** jak i w listach, które zostały rozesłane jako kopie **Cc**;

1.3.6 Pola referencyjne

Message-Id — Pole to zawiera unikalny (jeden dla każdej przesyłki, nigdy nie powtarzający się) identyfikator odnoszący się do bieżącej wersji tego listu. Unikalność identyfikatora gwarantowana jest przez hosta, który go tworzy. Specjalnie podkreślono słowa "bieżącej" i "tego", gdyż dla kolejnych wersji tego samego listu identyfikator powinien być różny.

In-Reply-To — Zawartość tego pola identyfikuje poprzedni list, na który obecny list jest odpowiedzią. W polu tym może występować identyfikator listu **Message-Id**.

References — Zawartość tego pola identyfikuje pozostałą korespondencję, do której odnosi się bieżący list. Podobnie jak i w polu **In-Reply-To**, w polu tym może występować identyfikator listu **Message-Id**.

Keywords — Pole to może zawierać słowa kluczowe lub frazy, oddzielone od siebie przecinkami.

1.3.7 Pozostałe pola

Subject — Pole to powinno zawierać krótki (najczęściej kilkuwyrazowy) opis zawartości lub natury listu.

Comments — Pole to pozwala na dołączenie komentarza do listu bez „zakłócania” (ingerowania w spójność treści) przesyłki.

Encrypted — W przypadku zaszyfrowania przesyłki, pole to powinno wskazywać oprogramowanie lub metodę użytą do zaszyfrowania wiadomości. Warto zauważyć, że nazwy oprogramowania używanego do szyfrowania wiadomości są rejestrowane przez NIC, SRI International, Menlo Park, California, USA.

1.4 Rekordy MX

Każdy komputer w sieci Internet ma przypisany unikalny adres tzw. adres IP. Adres taki to 32-bitowa liczba zapisana w konwencji **a.b.c.d**, gdzie **a.b.c.d** są liczbami z zakresu od 0 do 255. Na poziomie pojedynczych pakietów (czyli minimalnych paczek danych jakie można wysyłać w sieci) komputery wykorzystują właśnie adresy IP do komunikacji. Adresy te odzwierciedlają także

fizyczną strukturę sieci Internet. Na podstawie adresu komputer może stwierdzić czy odbiorca danego pakietu znajduje się w tej samej sieci fizycznej i można mu doręczyć dane bez żadnych pośredników, czy też trzeba przekazać je do innego „mądrzejszego” węzła sieci, który będzie wiedział, co dalej z takim pakietem zrobić.

Adresy IP są więc nieodzowne do funkcjonowania sieci w jej najniższych warstwach. Ale przesyłanie pakietów między komputerami już dawno przestało być sztuką dla sztuki, stało się środkiem służącym do komunikacji między ludźmi. A ludzie, nie są specjalnie biegli w zapamiętywaniu przypadkowych numerów. Za to zwykle wiedzą czego szukają, albo do kogo chcą wysłać list. I dlatego konieczne stało się nadawanie komputerom w sieci nie tylko adresów numerycznych ale także nazw tekstowych, które mówią coś np. na temat właściciela, położenia geograficznego czy przeznaczenia danej maszyny. Możliwe jest także nadawanie podobnych nazw maszynom, które mimo iż mogą być bardzo odległe geograficznie, to spełniają podobne funkcje, albo należą do tego samego właściciela.

System DNS (ang. *Domain Name System* czyli System Nazw Domen) [3] jest, mówiąc najkrócej, bazą danych służącą do odwzorowywania adresów tekstowych w numeryczne i odwrotnie. W tej bazie dostępne są także informacje nieco innego typu, np. jaka maszyna obsługuje pocztę dla danego ośrodka.

DNS używa jednego typu rekordu zasobów do realizacji rozszerzonego trasowania poczty - rekordu MX. Rekord MX określa *wymiennik poczty* (ang. *mail exchanger*) dla nazwy domeny: host, który będzie albo przetwarzał, albo przekazywał pocztę dla domeny (na przykład poprzez zaporę sieciową). *Przetwarzanie poczty* oznacza dostarczenie jej do osoby, do której jest adresowana, albo zbramkowanie w celu transportu innym protokołem. *Przekazywanie* (ang. *forwarding*) oznacza wysyłanie poczty do docelowego hosta albo do innego wymiennika poczty ”bliższego” celu, przez SMTP. Czasem przekazywanie poczty pociąga za sobą także umieszczenie wiadomości w kolejce na pewien czas.

Aby zapobiec pętłom w przekazywaniu poczty, rekord MX ma dodatkowy parametr poza nazwą domeny wymiennika poczty: *współczynnik preferencji*. Współczynnik preferencji jest 16-bitową liczbą bez znaku (pomiędzy 0 a 65535), która oznacza priorytet wymiennika poczty. Na przykład rekord MX:

```
math.uwb.edu.pl      IN      MX      10      mail.uwb.edu.pl
```

oznacza, że *mail.uwb.edu.pl* jest wymiennikiem poczty dla *math.uwb.edu.pl* o współczynniku preferencji 10.

Zebrane razem, współczynniki preferencji wymienników poczty oznaczają kolejność, w jakiej host obsługujący pocztę powinien ich używać. Sama wartość współczynnika preferencji nie jest ważna, jedynie jej stosunek do wartości w innych wymiennikach poczty: czy jest wyższa, czy niższa niż wartość współczynnika wymienników poczty tego hosta. Jeżeli jest więcej rekordów, to hosty obsługujące pocztę powinny próbować dostarczyć do wymienników poczty najpierw pocztę z najniższymi współczynnikami preferencji. Wyczuwa się tu

pewną sprzeczność - najbardziej preferowany wymiennik poczty ma najniższy współczynnik preferencji. Ale ponieważ współczynnik preferencji jest wartością bez znaku, pozwala to określić "najlepszy" wymiennik poczty współczynnikiem preferencji 0.

Jeśli dostawa do najbardziej preferowanego wymiennika poczty zawiedzie, oprogramowanie obsługujące pocztę powinno próbować dostarczyć ją mniej preferowanym wymiennikom poczty (tym z wyższymi wartościami współczynników preferencji), w kolejności wzrastających współczynników preferencji. Oznacza to, że oprogramowanie obsługujące pocztę powinno wypróbować bardziej preferowane wymienniki poczty przed mniej preferowanymi. Więcej niż jeden wymiennik poczty może mieć taki sam współczynnik preferencji, oprogramowaniu może więc ustalić kolejność wysyłania (np. najnowsza, ósma wersja sendmaila, wybiera losowo spośród wymienników poczty o tym samym współczynniku preferencji). Wszystkie wymienniki poczty o danym współczynniku preferencji powinny być wypróbowane przed przetwarzaniem następnej, wyższej wartości.

Na przykład rekordy MX dla *ora.com* mogłyby być takie:

<i>ora.com.</i>	IN	MX	0	<i>ora.ora.com.</i>
<i>ora.com.</i>	IN	MX	10	<i>ruby.ora.com.</i>
<i>ora.com.</i>	IN	MX	10	<i>opal.ora.com.</i>

Instruuja one oprogramowanie przesyłające pocztę, aby próbowało dostarczyć pocztę do *ora.com* przez wysłanie:

1. najpierw do *ora.ora.com*,
2. następnie do *ruby.ora.com* albo *opal.ora.com*,
3. na koniec do pozostałego wymiennika poczty o preferencji 10 (nie użytego w punkcie 2).

Oczywiście gdy oprogramowanie szczęśliwie dostarczy pocztę do jednego z wymienników poczty *ora.com*, może się zatrzymać. Po dostarczeniu poczty do *ora.com* nie potrzebuje podejmować prób z *ruby* ani z *opal*.

Co będzie, jeśli host nie ma żadnego rekordu MX? Czy oprogramowanie obsługi poczty po prostu nie dostarczy poczty do tego hosta? Rzeczywiście, możemy skompilować ostatnią wersję Sendmaila, która tak robi. Jednak większość dostawców tak kompiluje swoje wersje Sendmaili, by jeśli nie istnieje żaden rekord MX, co najmniej próbowały dostawy do adresu IP hosta [1].

Rozdział 2

Hosty wirtualne

2.1 Konta wirtualne

Dość często zdarza się sytuacja w firmie lub instytucji, która posiada własny serwer pocztowy, że część użytkowników potrzebuje wyłącznie skrzynki pocztowej, nie korzysta zaś z logowania się na serwer i nie wykonuje na nim żadnych programów. W tej sytuacji zamiast pełnego konta shellowego takiemu użytkownikowi wystarczy *konto wirtualne* [7]. Nie chodzi tu o ograniczenie możliwości użytkownika, ale o bezpieczeństwo serwera pocztowego. Zakładając konto shellowe zwiększamy ryzyko włamania na serwer. Dzięki kontom wirtualnym to ryzyko jest mniejsze, ponieważ *wirtualny użytkownik SMTP* to taki użytkownik, który nie ma żadnych praw, może wyłącznie odbierać i wysyłać pocztę. Konto wirtualne to nic innego jak pojedynczy plik na serwerze, w którym gromadzone są wiadomości przychodzące z Internetu. Plik ten często nazywa się *skrzynką wejściową* (ang. *Inbox*).

Nazwa tego pliku jest taka sama jak nazwa konta (nazwa użytkownika). Takie skrzynki wejściowe są pogrupowane w katalogach odpowiadających nazwom poszczególnych domen, ponieważ nazwa użytkownika może się powtórzyć w różnych domenach. Na przykład jeśli mamy następujących użytkowników: *jan@strona.com.pl*, *karol@strona.com.pl*, *jan@adres.com.pl*, to w katalogu *strona.com.pl* będą znajdowały się pliki *jan* oraz *karol*, zaś w katalogu *adres.com.pl* tylko plik *jan*.

Serwer SMTP (np. Sendmail) zapisuje w skrzynce wejściowej użytkownika przychodzące do niego wiadomości. Aby odczytać te wiadomości wirtualny użytkownik nie może użyć standardowego oprogramowania POP3 [5] czy IMAP [4], gdyż nie jest zarejestrowany jako użytkownik systemu. Jednym z programów, który pozwala na odbieranie poczty wirtualnemu użytkownikowi zgodnie z protokołem POP3 jest program *Teapop*. Jego podstawową zaletą jest to, że pozwala przechowywać informacje o wirtualnych domenach i kontach w bazie danych (w szczególności może to być baza MySQL).

Dane niezbędne do działania serwera Teapop zebrane są w bazie danych MySQL w tabeli *Mail*, której opis przedstawiono poniżej:

Field	Type	Null	Key	Default	Extra
ID	int(11)		PRI	NULL	auto_increment
TS	timestamp(14)	YES		NULL	
CT	timestamp(14)	YES		NULL	
username	varchar(16)		MUL		
domain	varchar(128)				
password	varchar(16)				
maildrop	varchar(16)	YES		NULL	
active	enum('T','N')	YES		T	
start_date	date			0000-00-00	
expire_date	date			0000-00-00	

Poszczególne kolumny tabeli *Mail* mają następujące znaczenie:

ID — unikalny identyfikator rekordu w tabeli,

TS — czas ostatniej modyfikacji rekordu,

CT — czas utworzenia rekordu,

username — nazwa konta użytkownika,

domain — nazwa domeny,

password — hasło użytkownika,

maildrop — względna ścieżka do skrzynki wejściowej użytkownika, liczona względem konfiguracji serwera Teapop,

active — pole decydujące o tym, czy dane konto jest aktywne (T), czy też zablokowane (N),

start_date — od kiedy konto jest aktywne,

expire_date — do kiedy konto jest aktywne.

Dla ułatwienia modyfikacji bazy danych przygotowany został wyspecjalizowany interfejs WWW, jako integralna część tej pracy. Aplikacja napisana jest języku Meta-HTML. Każdy użytkownik po zalogowaniu się do aplikacji ma możliwość zmiany swojego hasła. Zaś po zalogowaniu się przez administratora, ma on możliwość wglądu do wszystkich danych zawartych w bazie i modyfikowania ich.

2.2 Konfiguracja Sendmaila

Program Sendmail, podobnie jak inne tego typu programy na platformie Unix, działa w oparciu o swoje pliki konfiguracyjne. Są to zwykle pliki tekstowe (jak `sendmail.cf`; `sendmail.cw`) lub pliki w postaci bazy danych do ich szybkiego przetwarzania (`aliases`, `popauth`, `virtaliases`, `virtusertable`). Oto znaczenie poszczególnych plików konfiguracyjnych:

`access` to baza danych określająca możliwość dostępu do serwera Sendmail.

Określone są w niej hosty, które mają prawo wysyłać pocztę za pośrednictwem tego serwera oraz hosty z których poczta nie jest przyjmowana.

`aliases` to plik, który zawiera listę przydomków (aliasów).

`popauth` to dynamicznie generowana baza dostępu do serwera na podstawie autoryzacji użytkowników przez protokoły POP i IMAP.

`sendmail.cf` to główny plik konfiguracyjny serwera Sendmail.

`sendmail.cw` to plik w którym zawarta jest lista obsługiwanych przez serwer domen.

`virtaliases` to plik, który zawiera listę wirtualnych przydomków.

`virtusertable` jest to plik mapujący adresy wirtualne na adresy rzeczywiste lub aliasy do skrzynek wirtualnych. Każda linia pliku `virtusertable` zawiera adres wirtualny (który jest unikalny) i za nim oddzielony spacjami lub tabulacjami pełny adres zewnętrzny, nazwa konta lokalnego lub alias do skrzynki wirtualnej.

Podstawową rolę w konfiguracji hostów wirtualnych w Sendmailu odgrywają pliki `virtusertable` i `virtaliases`, jak same ich nazwy wskazują. Ponadto istotnym jest plik `sendmail.cw`.

Aby poczta wysłana do danej domeny docierała do naszego serwera w DNS-ie obsługujących tę domenę musi być ustawiony odpowiednio rekord MX. Na przykład jeśli naszym serwerem jest `theta.uwb.edu.pl`, zaś domenę której pocztę będziemy obsługiwać jest `solaris-x86.net`, a DNS-em obsługującym `solaris-x86.net` jest `math.uwb.edu.pl`, to na komputerze `math` musi być ustawiony następujący rekord MX:

```
solaris-x86.net      IN      MX      10      theta.uwb.edu.pl
```

Ponadto na komputerze `theta` w pliku `sendmail.cw` musi być umieszczony wpis `solaris-x86.net`, co spowoduje, że Sendmail na tym komputerze będzie przyjmował pocztę wysyłaną na tę domenę. Jeśli chcemy utworzyć wirtualnego użytkownika o nazwie np. `webmaster` w tej domenie, to po pierwsze musimy utworzyć plik, w którym będą gromadzone wiadomości wysyłane na adres: `webmaster@solaris-x86.net`. Powiedzmy, że plikiem tym jest:

```
/export/home1/mail/solaris-x86.net/webmaster
```

Program Sendmail musi mieć prawo zapisu do tego pliku. Następnie w pliku `virtusertable` umieszczamy wirtualny alias dla konta `webmaster`, tzn. wpisujemy:

```
webmaster@solaris-x86.net          solaris-x86.net__webmaster
```

Natomiast w pliku `virtaliases` dopisujemy

```
solaris-x86.net__webmaster: /export/home1/mail/solaris-x86.net/webmaster
```

Pamiętać należy o odpowiednim zawołaniu programu `makemap` po modyfikacji dwóch powyższych plików. Tak skonfigurowany Sendmail wstawi wiadomości wysłane na adres `webmaster@solaris-x86.net` do wskazanego pliku. To w jaki sposób Sendmail będzie traktował plik `virtusertable` i `sendmail.cw`, powiedziane jest w konfiguracji `sendmail.cf`. Aby w Sendmailu działały hosty wirtualne, plik na podstawie którego tworzymy `sendmail.cf`, musi zawierać:

```
FEATURE(virtusertable)
```

2.3 Odbieranie i wysyłanie poczty

Tak jak zostało powiedziane w 2.1, dobrym programem umożliwiającym użytkownikom odbieranie poczty jest Teapop. Jest to serwer działający zgodnie z protokołem POP3. Możemy więc łączyć się z nim przy pomocy standardowych programów do odbierania poczty (tj. Outlook Express, Netscape Messenger, Pegasus, itp.). Ponieważ nazwy użytkowników wirtualnych mogą się powtarzać w różnych domenach obsługiwanych przez jeden serwer, jako nazwę użytkownika podczas łączenia się z serwerem Teapop podajemy pełny adres e-mail (np. `webmaster@solaris-x86.net`), aby zapewnić jednoznaczną identyfikację z wirtualną skrzynką pocztową.

Niestety serwer pocztowy Teapop niewspółpracuje z protokołem szyfrowym SSL, więc przekazywane hasło do serwera nie jest szyfrowane i może być dość łatwo podejrzone w sieci. Ponieważ użytkownik wirtualny nie ma jednak żadnych praw na serwerze, to ewentualne włamanie może wyłącznie dotyczyć skrzynki wejściowej danego użytkownika. Jedyną szkodą jaką może wyrządzić haker, to przejrzanie lub skasowanie poczty wirtualnemu użytkownikowi.

Zazwyczaj serwerem SMTP służącym do wysyłania poczty jest ten sam serwer, który pocztę odbiera. W przypadku jednak dużych korporacji lub instytucji, ze względu na obciążenie serwera, odbieranie i wysyłanie poczty może być rozdzielone.

W przypadku wysyłania poczty istotne jest czy dany e-mail wysyłany jest w obrębie jednej domeny, czy też nie. W sytuacji, gdy poczta wysyłana jest na inną domenę, powstaje problem z tzw. *relaying*. Nadawca e-mail-a wysyłanego poza serwer SMTP powinien być tak zweryfikowany, aby uniemożliwić osobom

trzecim korzystanie z tego serwera. W przeciwnym razie serwer ten może być wykorzystany do rozsyłania tzw. SPAM-u lub anonimów. W dobrze skonfigurowanym programie Sendmail domyślnie nikt poza lokalnymi użytkownikami nie może wysyłać poczty. To, kto jest uprawniony do wysyłania poczty możemy kontrolować przy pomocy pliku konfiguracyjnego `access`. W pliku tym można podać adresy IP komputerów, z których pocztę można wysyłać. Można tam ponadto zablokować pocztę przychodzącą. Przykładowe wpisy w tym pliku wyglądają następująco:

```
212.33.73.244      RELAY
212.33.73.243      RELAY
acid.pl            REJECT
gambit.pl          REJECT
```

W powyższym przykładzie, użytkownicy komputerów o wymienionych adresach IP mogą wysyłać pocztę poza swoją domenę. Natomiast poczta przychodząca z domen *acid.pl* oraz *gambit.pl* jest odrzucana.

Powstaje poważny problem z użytkownikami serwera, którzy nie posiadają stałego adresu IP, np. gdy korzystają z modemu w domu, albo gdy adres IP jest adresem całej sieci, a nie pojedynczego komputera (tzn. jest adresem router'a, a nie komputera). Takim użytkownikom możemy zezwolić na wysyłanie poczty na dwa sposoby:

1. poprzez wystawienie użytkownikowi certyfikatu, na podstawie, którego serwer Sendmail dokonuje autoryzacji, albo
2. stosując oprogramowanie typu `popauth`, które dokonuje zmian w bazie `access` dynamicznie, na podstawie logowania się użytkowników poprzez usługi POP3 lub IMAP.

Program `popauth` działa na tej zasadzie, że sprawdza nieustannie dziennik systemowy `syslog` i wyłapuje z niego autoryzacje użytkowników dokonane podczas odbierania poczty. Sprawdza adres IP komputera, z którego nastąpiło połączenie i dopisuje go do bazy `popauth`, która ma to samo znaczenie co `access`. Sendmail traktuje równoważnie obie bazy. Po upływie ustalonego czasu (domyślnie pół godziny) wpis z bazy `popauth` jest usuwany i użytkownik traci możliwość wysyłania poczty poza swoją domenę.

Rozdział 3

Programy administracyjne

Aby ułatwić pracę Administratora w związku z wirtualnymi domenami i kontami, w ramach niniejszej pracy, opracowany został zestaw skryptów shellowych. Skrypty te przygotowane są do współpracy z serwerem Sendmail na platformie Solaris. Ze względu na użycie w nich wyłącznie podstawowych poleceń Unix, mogą one pracować na dowolnej platformie Unix lub Linux.

Programy zostały zainstalowane i uruchomione na komputerze `theta` w Instytucie Matematyki UwB oraz na kilku serwerach komercyjnych należących do prywatnych firm w Białymstoku.

3.1 Konfiguracja oprogramowania

Głównym zadaniem przygotowanych skryptów jest modyfikacja plików konfiguracyjnych Sendmaila. Aby ułatwić dostosowanie skryptów do różnych maszyn i platform, odwołanie do plików konfiguracyjnych zastąpiono zmiennymi środowiskowymi, których znaczenie i domyślne wartości zebrano poniżej.

`MAILROOTDIR` to ścieżka do katalogu, w którym przechowywane są wszystkie konta wirtualne, a dokładniej wszystkie katalogi, odpowiadające domenom wirtualnym, zawierające skrzynki wejściowe wirtualnych użytkowników.

Domyślna wartość `/export/home1/mail`

`MAKEMAP` to ścieżka do programu `makemap`.

Domyślna wartość `/usr/local/sbin/makemap`

`SENDMAILCW` to ścieżka do pliku konfiguracyjnego `sendmail.cw`.

Domyślna wartość `/opt/etc/mail/sendmail.cw`

`TEAPOPCONF` to ścieżka do pliku konfiguracyjnego programu `Teapop`.

Standardowo jest ona ustawiona jako `/opt/etc/teapop.passwd`

VIRTALIASES to ścieżka do pliku konfiguracyjnego `virtaliases`.

Domyślna wartość `/opt/etc/mail/virtaliases`

VIRTALIASADD to ścieżka do skryptu `virtaliasadd`.

Domyślna wartość `/opt/sbin/virtaliasadd`

VIRTALIASDEL to ścieżka do skryptu `virtaliasdel`.

Domyślna wartość `/opt/sbin/virtaliasdel`

VIRTUSERTABLE to ścieżka do pliku konfiguracyjnego `virtusertable`.

Domyślna wartość `/opt/etc/mail/virtusertable`.

Jeżeli modyfikowane są pliki konfiguracyjne Sendmaila, które wymagają przetworzenia do postaci binarnej bazy danych, to wtedy wołany jest program `makemap` z odpowiednimi argumentami. Aby dokonane zmiany w konfiguracji Sendmaila zaczęły obowiązywać, należy polecić Sendmailowi ponowne przeczytanie swoich plików konfiguracyjnych. Wykonuje się to przesyłając sygnał HUP do procesu Sendmaila np.

```
pkill-HUP sendmail
```

Nie przerywa to pracy programu, a więc i przetwarzania poczty w tym czasie.

3.2 Obsługa wirtualnych domen

Nazewnictwo skryptów zostało zapożyczony od zestawu standardowych programów do zarządzania kontami użytkowników na platformie Solaris, a mianowicie `useradd`, `usermod`, `userdel` etc.

3.2.1 Dodanie nowej domeny

Program `virtdomainadd` dodaje nową wirtualną domenę. Uruchamia się go podając jako argument nazwę tworzonej domeny. Zostaje ona dopisana do pliku `sendmail.cw`, gdzie umieszczone są wszystkie obsługiwane przez serwer domeny. Program zaczyna od sprawdzenia czy istnieje kartoteka, w której przechowywane będą wszystkie skrzynki wirtualne należące do danej domeny. Jeżeli jej nie ma to tworzona jest taka kartoteka. Kartoteka domeny i wszystkie pliki w tej kartotece muszą mieć ustawione prawo zapisu dla grupy `mail`, tak aby umożliwić zapis do skrzynek wirtualnych przez program Sendmail. Dodawany jest odpowiedni wpis do konfiguracji Teapop, tak by umożliwić właścicielom kont wirtualnych odbieranie poczty. Serwer Teapop jest restartowany, aby zmiany zostały uaktywnione.

Kod źródłowy na str. 22.

3.2.2 Usunięcie domeny

Aby usunąć wirtualną domenę należy wywołać program `virtomaindel` podając nazwy domen do usunięcia jako argumenty. Może być ich więcej niż jedna.

Program usuwa domenę z pliku `sendmail.cw`. Usuwane są również wszystkie aliasy związane z tą domeną w plikach `virtusertable` i `virtaliases`. Następnie usuwany jest katalog, w którym przechowywane są wirtualne skrzynki tej domeny, modyfikowana jest konfiguracja programu Teapop i usuwana jest obsługa tej domeny, po czym program Teapop jest restartowany, aby uaktywnić zmiany.

Kod źródłowy na str. 24.

3.3 Obsługa wirtualnych kont

3.3.1 Założenie konta wirtualnego użytkownika

Do zakładania wirtualnych kont służy program `virtuseradd`. Uruchamia się go podając jako argument pełny adres e-mail, czyli adres użytkownika i domenę rozdzielone znakiem `@`.

Program `virtuseradd` dodaje do pliku `virtusertable` nowy wiersz zgodnie z budową tego pliku. Przed wykonaniem zmian w pliku `virtusertable` sprawdzany jest czy ten plik w ogóle istnieje i czy domena dopisywanego adresu wirtualnego jest obsługiwana przez nasz serwer SMTP. Sprawdzenie to polega na porównaniu części domenowej podanego adresu wirtualnego z zawartością pliku `sendmail.cw`, który zawiera listę obsługiwanych domen. Następnie sprawdzane jest czy skrzynka wejściowa użytkownika istnieje i jeżeli jej nie ma to tworzony jest pusty plik, w którym gromadzone będą wiadomości przychodzące do tego użytkownika. Plik ten będziemy nazywać *wirtualną skrzynką pocztową*.

Kod źródłowy na str. 28.

3.3.2 Usunięcie konta wirtualnego użytkownika

Do usunięcia konta wirtualnego użytkownika służy program `virtuserdel`. Konta, które chcemy usunąć wypisujemy jako argumenty tego programu. Program `virtuserdel` wykonuje operacje odwrotną do `virtuseradd` tzn. usuwa wpisy w plikach `virtaliases` i `virtusertable` oraz usuwa skrzynkę wskazanego użytkownika. Po zakończeniu programu `virtuserdel` system jest w takim stanie jak przed wykonaniem odpowiedniego polecenia `virtuseradd`.

Kod źródłowy na str. 30.

3.3.3 Utworzenie wirtualnego aliasu

Program `virtaliasadd` dodaje do pliku `virtusertable` aliasy do kont pocztowych. Jako pierwszy parametr podajemy pełny adres wirtualny, a jako drugi adres rzeczywisty lub nazwę konta lokalnego.

Dodawane aliasy mają dwa różne znaczenia. Pierwsza grupa aliasów to aliasy do wirtualnych kont pocztowych na tym serwerze. Natomiast druga grupa to aliasy do kont pocztowych znajdujących się na innych serwerach. O ile pierwsze wymagane są ze względu na funkcjonowanie wirtualnych kont pocztowych, to drugie służą do przekazania poczty innym serwerom, na których znajdują się odpowiednie konta pocztowe.

Kod źródłowy na str. 26.

3.3.4 Usunięcie wirtualnego aliasu

Do usuwania wirtualnego aliasu używany jest program `virtaliasdel` z podaną listą pełnych adresów wirtualnych do usunięcia. Program `virtaliasdel` usuwa z pliku `virtusertable` podane adresy wirtualne. Jest to zadanie odwrotne do tego co wykonuje program `virtaliasadd`.

Kod źródłowy na str. 27.

Dodatek A

Skrypty

virtdomainadd

```
#!/bin/sh
#
# Add virtual mail domain.
#
# Last modified: May 23, 2003
#
# Copyright (c) 2003 Pawel Kiersnowski & Mariusz Zynel.
#
# This software is FREE. You can use and/or redistribute it for any
# purpose in either, modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission notice shall be included
# in all copies or substantial portions of this software.
#
# THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
# EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

SELF='basename $0'

PATH=/usr/bin:/usr/sbin
export PATH

SENDMAILCW=/opt/etc/mail/sendmail.cw
MAILROOTDIR=/export/home1/mail
TEAPOPCONF=/opt/etc/teapop.passwd

DOMAIN=$1
DOMAINDIR=${MAILROOTDIR}/${DOMAIN}

if [ $# -ne 1 ]; then
    echo "Usage: _$SELF_ domain"
    exit 1;
fi

# Check if the domain already exists
if [ -n "$(grep _$DOMAIN_ $SENDMAILCW)" ]; then
    echo "ERROR: _$DOMAIN_ already exists in _$SENDMAILCW_"
    exit 2;
fi

# Add domain to sendmail configuration
```

```
echo $DOMAIN >> $SENDMAILCW

# Check if the domain directory exists and create it
if [ ! -d $DOMAINDIR ]; then
    mkdir -p $DOMAINDIR
fi
chgrp mail $DOMAINDIR
chmod 0775 $DOMAINDIR

# Add the domain to TEAPOP configuration
if [ -f $TEAPOPCONF ]; then
    echo "$DOMAIN:*:mysql:$MAILROOTDIR/$DOMAIN:1:root:mail:localhost:\
dbISP:teapopmgr:t3aPOP:tMail:username:password:maildrop:domain:" _\
    >> $TEAPOPCONF
    /etc/init.d/teapop stop
    /etc/init.d/teapop start
fi

# Make Sendmail to reread config files
kill -HUP sendmail
```

virtomaindel

```
#!/bin/sh
#
# Remove virtual mail domain.
#
# Last modified: May 23, 2003
#
# Copyright (c) 2003 Pawel Kiersnowski & Mariusz Zynel.
#
# This software is FREE. You can use and/or redistribute it for any
# purpose in either, modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission notice shall be included
# in all copies or substantial portions of this software.
#
# THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
# EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

SELF='basename $0'

PATH=/usr/bin:/usr/sbin
export PATH

SENDMAILCW=/opt/etc/mail/sendmail.cw
MAILROOTDIR=/export/home1/mail
VIRTUSERTABLE=/opt/etc/mail/virtusertable
VIRTALIASES=/opt/etc/mail/virtaliases
TEAPOPCONF=/opt/etc/teapop.passwd
MAKEMAP=/usr/local/sbin/makemap

if [ $# -lt 1 ]; then
    echo "Usage: _$SELF_ domain _..."
    exit 1;
fi

# Delete the domain from sendmail.cw file
if [ -f $SENDMAILCW ]; then
    for d in $*; do
        sed "/^{d}$/" $SENDMAILCW > $TMPFILE
        mv $TMPFILE $SENDMAILCW
    done
else
    echo "ERROR: _$SENDMAILCW_ doesn't exist"
fi

# Delete from virtusertable all virtual aliases tha belong to the domain
if [ -f $VIRTUSERTABLE ]; then
    for d in $*; do
        sed "/^[^_@]@$d[_]/" $VIRTUSERTABLE > $TMPFILE
        mv $TMPFILE $VIRTUSERTABLE
    done
    $MAKEMAP hash $VIRTUSERTABLE < $VIRTUSERTABLE
else
    echo "ERROR: _$VIRTUSERTABLE_ doesn't exist"
fi

# Delete from virtaliases all virtual aliases tha belong to the domain
if [ -f $VIRTALIASES ]; then
    for d in $*; do
        sed "/^{d} _/" $VIRTALIASES > $TMPFILE
        mv $TMPFILE $VIRTALIASES
    done
fi
```

```
$MAKEMAP hash $VIRTALIASES < $VIRTALIASES
else
  echo "ERROR: _$VIRTALIASES_ doesn't _exist"
fi

# Remove domain directory
for d in $*; do
  rm -rf ${MAILROOTDIR}/${d}
done

# Remove the domain from TEAPOP configuration
if [ -f $TEAPOPCONF ]; then
  for d in $*; do
    sed "/^${d}:/d" $TEAPOPCONF > $TMPFILE
    mv $TMPFILE $TEAPOPCONF
  done
  /etc/init.d/teapop stop
  /etc/init.d/teapop start
fi

# Make Sendmail to reread config files
pkill -HUP sendmail
```

virtaliasadd

```
#!/bin/sh
#
# Add virtual mail alias.
#
# Last modified: May 23, 2003
#
# Copyright (c) 2003 Pawel Kiersnowski & Mariusz Zynel.
#
# This software is FREE. You can use and/or redistribute it for any
# purpose in either, modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission notice shall be included
# in all copies or substantial portions of this software.
#
# THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
# EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

SELF='basename $0'

PATH=/usr/bin:/usr/sbin
export PATH

SENDMAILCW=/opt/etc/mail/sendmail.cw
VIRTUSERTABLE=/opt/etc/mail/virtusertable
MAKEMAP=/usr/local/sbin/makemap

if [ $# -ne 2 ]; then
    echo "Usage: _$SELF_ username@domain _existing_address"
    exit 1;
fi

# Check if it is a mail within our domains
DOMAIN='echo $1 | sed 's/.*@//''
if [ -z "$(grep _$DOMAIN_ $SENDMAILCW)" ]; then
    echo "ERROR: _$DOMAIN_ is not in _$SENDMAILCW"
    exit 2;
fi

# Add virtual address if it doesn't yet exist
if [ -f $VIRTUSERTABLE ]; then
    if [ -z "$(awk '{print _$1_}' _$VIRTUSERTABLE_ | grep _$1'" ); then
        echo "$1\t\t$2" >> $VIRTUSERTABLE
        $MAKEMAP hash $VIRTUSERTABLE < $VIRTUSERTABLE
    else
        echo "ERROR: _$1_ already _exists"
    fi
else
    echo "ERROR: _$VIRTUSERTABLE_ doesn't _exist"
fi

# Make Sendmail to reread config files
pkill -HUP sendmail
```

virtaliasdel

```
#!/bin/sh
#
# Remove virtual mail alias.
#
# Last modified: May 23, 2003
#
# Copyright (c) 2003 Pawel Kiersnowski & Mariusz Zynel.
#
# This software is FREE. You can use and/or redistribute it for any
# purpose in either, modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission notice shall be included
# in all copies or substantial portions of this software.
#
# THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
# EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

SELF='basename $0'

PATH=/usr/bin:/usr/sbin
export PATH

VIRTUSERTABLE=/opt/etc/mail/virtusertable
MAKEMAP=/usr/local/sbin/makemap

TMPFILE=/tmp/${SELF}.tmp

if [ $# -lt 1 ]; then
    echo "Usage: _$SELF_ username@domain _..."
    exit 1;
fi

# Delete virtual address from the table
if [ -f $VIRTUSERTABLE ]; then
    for v in $*; do
        sed "/^$v[_-]/d" $VIRTUSERTABLE > $TMPFILE
        mv $TMPFILE $VIRTUSERTABLE
    done
    $MAKEMAP hash $VIRTUSERTABLE < $VIRTUSERTABLE
else
    echo "ERROR: _$VIRTUSERTABLE_ doesn't exist"
fi

# Make Sendmail to reread config files
pkill -HUP sendmail
```

virtuseradd

```
#!/bin/sh
#
# Add virtual mail account.
#
# Last modified: May 23, 2003
#
# Copyright (c) 2003 Pawel Kiersnowski & Mariusz Zynel.
#
# This software is FREE. You can use and/or redistribute it for any
# purpose in either, modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission notice shall be included
# in all copies or substantial portions of this software.
#
# THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
# EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

SELF='basename $0'

PATH=/usr/bin:/usr/sbin
export PATH

SENDMAILCW=/opt/etc/mail/sendmail.cw
VIRTUSERTABLE=/opt/etc/mail/virtusertable
VIRTALIASES=/opt/etc/mail/virtaliases
MAILROOTDIR=/export/home1/mail
MAKEMAP=/usr/local/sbin/makemap
VIRTALIASADD=/opt/sbin/virtaliasadd

if [ $# -ne 1 ]; then
    echo "Usage: _$SELF_ username@domain"
    exit 1;
fi

USERNAME='echo $1 | sed 's/@.*//''
DOMAIN='echo $1 | sed 's/.*@//''
DOMAINDIR=${MAILROOTDIR}/$DOMAIN
INBOX=${DOMAINDIR}/$USERNAME

# Check if it is a mail within our domains
if [ -z "$(grep _$DOMAIN_ $SENDMAILCW)" ]; then
    echo "ERROR: _$DOMAIN_ is not in _$SENDMAILCW"
    exit 2;
fi

# Exit if the inbox already exists
if [ -f $INBOX ]; then
    echo "ERROR: _${USERNAME}@$DOMAIN_ already exists"
    exit 3;
fi

# Create virtual inbox
touch $INBOX
chgrp mail $INBOX
chmod 0660 $INBOX

# Add user address to virtusertable
$VIRTALIASADD ${USERNAME}@$DOMAIN ${DOMAIN}_$USERNAME

# Add virtual address if it doesn't yet exist
if [ -f $VIRTALIASES ]; then
```

```
if [ -z "`awk '{print $1}' $VIRTALIASES | grep $USERNAME`" ]; then
    echo "${DOMAIN}_$USERNAME:\t\t$INBOX" >> $VIRTALIASES
    $MAKEMAP hash $VIRTALIASES < $VIRTALIASES
else
    echo "ERROR: $USERNAME already exists"
fi
else
    echo "ERROR: $VIRTALIASES doesn't exist"
fi

# Make Sendmail to reread config files
pkill -HUP sendmail
```


virtuserdel

```
#!/bin/sh
#
# Remove virtual mail account.
#
# Last modified: May 23, 2003
#
# Copyright (c) 2003 Pawel Kiersnowski & Mariusz Zynel.
#
# This software is FREE. You can use and/or redistribute it for any
# purpose in either, modified, or unmodified form, under the terms of the
# GNU General Public License as published by the Free Software Foundation.
#
# The above copyright notice and this permission notice shall be included
# in all copies or substantial portions of this software.
#
# THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
# EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
# LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

SELF='basename $0'

PATH=/usr/bin:/usr/sbin
export PATH

VIRTUSERTABLE=/opt/etc/mail/virtusertable
VIRTALIASES=/opt/etc/mail/virtaliases
MAILROOTDIR=/export/home1/mail
VIRTALIASDEL=/opt/sbin/virtaliasdel

TMPFILE=/tmp/${SELF}.tmp

if [ $# -lt 1 ]; then
    echo "Usage: _$SELF_ username@domain _..."
    exit 1;
fi

# Delete virtual address from virtusertable
if [ -f $VIRTUSERTABLE ]; then
    for v in $*; do
        USERNAME='echo $1 | sed 's/@.*//''
        DOMAIN='echo $1 | sed 's/.*@//''
        DOMAINDIR=$MAILROOTDIR/$DOMAIN
        INBOX=$DOMAINDIR/$USERNAME
        sed "/^$DOMAIN_$USERNAME:/d" $VIRTALIASES > $TMPFILE
        mv $TMPFILE $VIRTALIASES
        $VIRTALIASDEL $v
        rm -f $INBOX
    done
    $MAKEMAP hash $VIRTALIASES < $VIRTALIASES
else
    echo "ERROR: _$VIRTUSERTABLE_ doesn't exist"
fi

# Make Sendmail to reread config files
pkill -HUP sendmail
```

Spis literatury

- [1] Paul Albitz, Cricket Liu, *DNS and BIND*, 4th Edition, O'Reilly, 2001.
- [2] Bryan Costales, Eric Allman, *Sendmail*, O'Reilly, 1997.
- [3] DNS Resources Directory
<http://www.dns.net/dnsrd/>
- [4] Internet Message Access Protocol - Version 4rev1,
<http://www.faqs.org/rfcs/rfc3501.html>
- [5] Post Office Protocol - Version 3,
<http://www.faqs.org/rfcs/rfc1939.html>
- [6] Simple Mail Transfer Protocol
<http://www.faqs.org/rfcs/rfc821.html>
- [7] Virtual Hosting with Sendmail
<http://www.sendmail.org/virtual-hosting.html>