

Solutions to the Internet UCE Problem

Blue Reef Consulting, Inc.

<http://www.blureef.net>

Revised: 2/24/98

1.0 Introduction

First, let's define what is UCE (Unsolicited Commercial Email) or spam. One of the best definitions comes from <http://spam.abuse.net/whatisspam.html>:

"Spam [UCE] is flooding the Internet with many copies of the same message, in an attempt to force the message on people who would not otherwise choose to receive it. Most spam is commercial advertising, often for dubious products, get-rich-quick schemes, or quasi-legal services. Spam costs the sender very little to send -- most of the costs are paid for by the recipient or the carriers rather than by the sender."

As an ISP/Reseller you should adopt and enforce a spam or UCE policy to be a good "Netizen". Make sure you publish your policy to your clients and potential clients. Many good anti-spam policies, including Blue Reef's, can be found at <http://spam.abuse.net/goodsites/index.html>. You should also show your support for the proposed law to outlaw spam on the Internet (H.R. 1748, "The Netizens Protection Act of 1997") by writing/calling your congressman. For more information about this see <http://www.cauce.org/>.

Until there is a law to outlaw spam there are a few things you can do now to make the spammer's life tough. Before we describe possible solutions, let's make sure we understand the problem. There are many ways that a spammer may abuse your server, including:

1. Send spam or UCE to your or your customers' email boxes.
2. Relay spam or UCE through your server in attempt to mask their true identity.
3. Send spam or UCE to users on the Internet using (or spoofing) your domain as the return address. There is really nothing that can be done here short of legal action¹.

As for the first two, there are methods to block spam from spammers and block spammers from using your server as a relay host. The rest of this paper describes in detail some ways to do both of these.

2.0 Blocking Spam

1. At least one company is in the process of doing this. See <http://spam.abuse.net/news/typhoon.txt>.

2.1 “Mail From” Address Filtering

There are a number of ways to block email from spammers. Blue Reef Consulting has included with all accounts an easy and straight forward way to block email from an address or domain name. In your `~/etc/` directory there is a file called "spammers" which contains something like the following:

```
#
# Simply add the hostnames or email addresses on their own lines below
# Examples:
# cyberpromo.con (blocks all mail from the hostname "cyberpromo.con")
# spammer@aol.con (blocks mail only from "spammer@aol.con")
#
spammer@aol.con
cyberpromo.con
```

Lines that begin with "#" are comments. Other lines are domains or email addresses to block. In the above example, all email with the email address "spammer@aol.con" or any email address with the hostname "cyberpromo.con" in the "From:" line is blocked by the server. To add additional email address or domain names to block, do the following from the TELNET prompt of your Virtual Server:

1. Edit your `~/etc/spammers` file:

```
% pico ~/etc/spammers
      <add the domain name or email address at the end>
```

2. Run the "vnewspammers" to update the spammer database:

```
% vnewspammers
```

Important Note: If your account has been with Blue Reef for a long time (or you are missing the "`~/etc/spammers`" file) you may need to run "updatesendmailcf" command at the TELNET prompt to add this Anti-Spam feature.

For well known spammer's email addresses and domains for your `~/etc/spammers` file see:

```
http://www.webeasy.com:8080/spam/spam_download_table
http://www.idot.aol.com/preferredmail/
http://www.cco.caltech.edu/~cbrown/BL/
http://www-math.uni-paderborn.de/%7Eaxel/BL/
```

or

send email to bobby@fujipub.com (Bobby Holstein) with the subject line "SPAMMERFILE"

or

send email to spamlist@us.net

The problem with the above solution is that spammers like to forge their addresses and try to mask their identity and origin in an attempt to circumvent this type of block. This can be easily done by the spammer by forging the “Mail From:” line in SMTP (Simple Mail Transfer Protocol). Therefore, you may have to filter on other things in the headers.

2.2 “procmail” Filtering

The procmail mail filtering program, developed by Stephen van den Berg at RWTH-Aachen, Germany, provides a powerful mechanism to analyze and filter incoming mail for anything in the headers or body. The most recent version of procmail is always available for free download from

<ftp://ftp.informatik.rwth-aachen.de/pub/packages/procmail>

While an occasional spam message might slip through even the most careful defenses, taking steps to actively block spammers will greatly reduce the numbers of spam messages delivered to your users by your virtual server.

Because procmail is well-suited to processing mail messages, it forms a handy weapon in the war against spam. Catherine Hampton has developed a set of procmail configuration files, named the “Spam Bouncer”, that can be used to implement a sophisticated spam blocking scheme with procmail. The latest version of the Spam Bouncer is always available for free download from

<http://www.best.com/~ariel/nospam/index.shtml>

She has also prepared a set of instructions for new procmail users interested in understanding how the Spam Bouncer and procmail interact, available at

<http://www.best.com/~ariel/nospam/proctut.shtml>

One of the interesting capabilities of the Spam Bouncer is its ability to divide potentially objectionable mail into two different levels of suspicion: blatant spam messages and questionable messages. Messages that are determined to be blatant spam may be discarded, archived for later review, or even may be made the object of an automated complaint filed with the connectivity providers of some of the most notorious spammers. If the message is deemed to be questionable, either because it came from a site known to harbor spammers or it contains some of the catch phrases commonly used by spammers in their messages, it can be returned to the sender with a message including a keyword to be used in the header of future messages. Since spammers rarely use valid return addresses in their mail, and since the likelihood that they personally read their replies is small, this provides a mechanism for selectively accepting messages that would otherwise be rejected. Future messages containing the required keyword are automatically accepted by the Spam Bouncer, thus permitting safe delivery of the message.

Installing, configuring, and maintaining the Spam Bouncer are tasks recommended only for advanced users since the procedure requires a moderate level of UNIX knowledge and a solid understanding of the virtual server environment. The Spam Bouncer documentation pages mentioned above provide thorough documentation and should be consulted regularly since the Spam Bouncer is often updated and improved.

3.0 Blocking Relaying

Other spammers, rather than sending their abusive email to you and your customers, may use your Virtual Server's email server to "relay" their spam to other people. This is an attempt to make it look like it was your site that sent UCE. Here is what can be done to stop abusive relaying through your server:

1. Add information in your email headers to help the victim identify the true spammer or sender of the email.

2. Shut down all relaying except for those sites on the Internet that you approve. In other words, only IP addresses or domain names in your "access control list" will be able to use your resources.
3. Shut down relaying for those that prove to be abusive. In other words, everyone except those in your access control list will be able to use your resources.
4. Require your users to authenticate with the server before they can relay email.

3.1 Email Header Modification

We have two suggestions for adding to your email headers. The first is turning on what is called authentication-warnings or "authwarnings". Setting the authwarnings option causes your email server to insert special headers into the mail message that advise the recipient of reasons to suspect that the message may not be authentic. When this is set and email is "spoofed" (forged through your site) the recipient will see something like the following in their email headers:

```
X-Authentication-Warning: yourdomain.com: Host spammer.cyberpromo.com
[192.168.29.130] claimed to be yourfriend.com
```

Authwarnings are added by editing your `~/etc/sendmail.cf` file and adding the following:

```
Opauthwarnings
```

to the "options" area. Here is how you do this:

1. Using your favorite UNIX editor, edit `~/etc/sendmail.cf`

```
% pico ~/etc/sendmail.cf
```

Look for the area in the `sendmail.cf` file that looks like:

```
#####
# Options #
#####

Oa1                # Wait (in minutes) for alias file rebuild
OA/etc/aliases     # location of alias file
OC10               # Checkpoint queue runs every N deliveries
OF0600            # Temporary file mode
Og100             # Default GID
OH/etc/sendmail.hf # SMTP help file
OI                # Insist that the name server be running
Ok5               # Open connection cache size
Om               # Expand aliases to include sender
On               # Verify RHS in newaliases
OQ/usr/spool/mqueue # Queue directory
OS/etc/sendmail.st # Stat file
OT3d             # Queue timeout and warning time
Ot               # Use TZ environment variable
```

If you already have a line in the options area that begins with an "Op" then just add "authwarnings" to that line, separated from the other flags with a comma. If you don't already have an "Op" line, then add the following below that last option line (lines that begin with "O"):

```
Opauthwarnings    # Privacy Options: Authentication Warnings
```

Another suggestion is to add information about who to contact if there is a problem with mail coming from your server. For example, something like the following could be added about email "passing through" your server:

```

X-Info1: *****
X-Info2: * This email came through the SMTP.ABC.ORG email server. If you *
X-Info3: * suspect this email was sent by a spammer through this site *
X-Info4: * please forward the ENTIRE email message including headers to *
X-Info5: * abuse@abc.org so action can be taken against the spammer. *
X-Info6: **** Fight Spam on the Internet! See http://spam.abuse.net ****
X-Info7: **** Outlaw Spam: Support HR 1748 See http://www.cauce.org ****
X-Info8: **** Block Spam: Blackhole list see http://maps.vix.com/rbl ****
X-Info9: *****

```

This educates the recipient that the email did indeed come from your server and you are willing to do something about the message if they are willing to forward you the message. Note that spammers will also see this message when they send a few test messages to see what your headers look like. With the right message, like above, they may think twice about using your server for their UCE.

Here is how you add your personal header message in your email:

1. With your favorite UNIX editor, edit your `~/etc/sendmail.cf` file:

```
% pico ~/etc/sendmail.cf
```

In the "Header formats" section of file, and after the very last line that begins with "H", add something like the following:

```

H?M?X-Info1: $.*****
H?M?X-Info2: $.* This email came through the SMTP.ABC.ORG email server. If you*
H?M?X-Info3: $.* suspect this email was sent by a spammer through this site *
H?M?X-Info4: $.* please forward the ENTIRE email message including headers to *
H?M?X-Info5: $.* abuse@abc.org so action can be taken against the spammer. *
H?M?X-Info6: $.*** Fight Spam on the Internet! See http://spam.abuse.net ***
H?M?X-Info7: $.*** Outlaw Spam: Support HR 1748 See http://www.cauce.org ***
H?M?X-Info8: $.*** Block Spam: Blackhole list See http://maps.vix.com/rbl ***
H?M?X-Info9: $.*****

```

This will place the information lines below the lines in the header.

3.2 Relay Blocking Except for Approved Sites

Rather than letting anyone relay email through your site, you may want to block relaying except for sites that have your approval. Again, this requires you to edit your `~/etc/sendmail.cf` file to add an additional "rule".

1. Like above add the following to your `sendmail.cf` file (just before the "Ruleset 0" area) using your favorite UNIX editor replacing "[tab]" with the tab key:

```

#####
# Ruleset check_rcpt - Shutdown relaying through this server #
#####

# dequoting map - Needed for SPAM hack below
Kdequote dequote

# permitted relay sites file

F{RelayOK} -o /etc/relayok.txt

Scheck_rcpt
# anything terminating locally is ok
R<$+ @ $=w >[tab]          $@ OK
R<$+ @ $* $={RelayOK} >[tab]  $@ OK

```

```

# anything originating locally is ok
R$*[tab]           $: $(dequote "" ${client_name} $)
R$=w[tab]         $@ OK
R$* ${RelayOK}[tab] $@ OK
R$@[tab]         $@ OK
# anything else is bogus
R$*[tab]         $#error $: "550 Relaying Denied"

```

IMPORTANT NOTE: [tab] should be replaced with tab's. This new rule ("check_rcpt") will not work without tab's.

2. Create the file ~/etc/relayok.txt and add all the domain names or hostnames that are allowed to relay mail through your site; one per line. For example,

```

bluereef.net
mycomputer.mydomain.com
myisp.com

```

Make sure you don't have blank lines in this file.

One problem with the above "anti-relaying" rule is that it relies on the DNS "reverse lookup" to be correct. Spammers love to spoof their reverse lookup. In other words, the spammer could guess one of the domain names in your ~/etc/relaydomains.txt list (like your domain name) and use it. However, it is very difficult to spoof IP addresses to the point that it is not worth the spammer's time. Therefore, to relay by IP addresses, you can use the following rule instead of the one above:

```

#####
# Ruleset check_rcpt - Shutdown relaying through this server #
#####

# dequoting map - Needed for SPAM hack below
Kdequote dequote

# permitted relay sites file

F{RelayOK} -o /etc/relayok.txt

Scheck_rcpt
# Anything terminating locally is ok
R< $+ @ $=w >[tab]      $@ OK
# Anything originating locally is ok
R$*[tab]               $: $(dequote "" ${client_name} $)
R$=w[tab]             $@ OK
R$@[tab]              $@ OK
# Check the client address
R$+[tab]              $: $(dequote "" ${client_addr} $) $| $1
RO $| $*[tab]         $@ ok
R$={RelayOK}$* $| $*[tab] $@ ok
# anything else is bogus
R$*[tab]              $#error $: "550 Relaying Denied"

```

Again, remember to replace the "[tab]" with the tab character.

In this case, the ~/etc/relayok.txt file should look something like:

```

127.0.0.1
192.168.1.100
10.123

```

Again, don't include any blank lines. The first line ("127.0.0.1") you should always have. This is a reserved IP address for "localhost". The second line shows how you can allow the host at 192.168.1.100 to

relay email through your site. The third line is an example of allowing a whole subnet relay through your site. For example, this could be the subnet for your ISP that you use. In other words, when you dialup to your ISP you always get an address that starts with "10.123" but the last two numbers are different every time. Note that you should include the IP address of your Virtual Server so email can be sent from it.

3.3 Relay Blocking for Spam Sites Only

Now, rather than blocking all IP addresses except the ones listed in relayok.txt let's say you want to allow all IP addresses except the ones listed in the file relaynotok.txt. This can be done by modifying the previous example (in section 3.2):

1. Change the following lines from

```
# permitted relay sites file
F{RelayOK} -o /etc/relayok.txt
```

to:

```
# not permitted relay sites file
F{RelayNotOK} -o /etc/relaynotok.txt
```

2. Now, modify the check_rcpt rule a little from:

```
R$={RelayOK}$* $| $*[tab]    $@ ok
# anything else is bogus
R$*[tab]                      $#error $: "550 Relaying Denied"
```

to:

```
R$={RelayNotOK}$* $| $*[tab]    $#error $: "550 Relaying Denied"
# anything else is OK
R$*[tab]                          $@ ok
```

Again, [tab]'s should be replaced with actual tab characters.

With these modifications if you detect someone relaying through your virtual server you can simply add the spammer's IP address block in your ~/etc/relaynotok.txt file. For example, say someone complains to you about spam that is coming from your site (yourdomain.com). From the headers of the spammer's email you should see something like the following:

```
Received: from yourfriend.com (smtp.spammer.con [192.168.1.232]) by yourdomain.com
(8.8.5) id NAA29638; Thu, 9 Oct 1997 13:32:33 -0600 (MDT)
Date: Thu, 9 Oct 1997 13:32:33 -0600 (MDT)
```

Also, in your ~/usr/log/messages file you should see something like:

```
<22>Oct 9 13:33:37 sendmail[29638]: NAA29638: from=<happy@your
friend.com>, size=5, class=0, pri=30005, nrcpts=1,
msgid=<199710091932.NAA29638@yourdomain.com>, proto=SMTP,
relay=smtp.spammer.con [192.168.1.232]
```

If you have the authwarnings set in your sendmail.cf's options (see section 3.1) you may also see the following in your ~/usr/log/messages file:

```
<22>Oct 9 13:33:37 sendmail[29638]: [NOQUEUE]: Authentication-Warning:
```

```
yourdomain.com: Host smtp.spammer.com [192.168.1.232] claimed to be yourfriend.com
```

In this example above the spammer is spoofing his domain to be yourfriend.com. However, from the email headers and log files you know the true IP address he is using: 192.168.1.232. To block this spammer from using your Virtual Server as a relay host for his UCE simply add this IP address (192.168.1.232) or his whole IP block by adding (192.168.1) to your `~/etc/relaynotok.txt` file.

3.4 Relay Blocking for Everyone Except Authenticated Email Users

Another approach is to shutdown all relaying except for users that can prove their identity through password authentication. Unfortunately, SMTP (Simple Mail Transfer Protocol) currently does not include authentication. While it has been proposed to add authentication within to SMTP this idea is still in the draft stage and is not implemented in any known clients at this time (See [11]). Also, it has been proposed that POP (Post Office Protocol) be extended to allow clients to send mail through the POP3 session rather than using SMTP. This works well since POP already requires the user to authenticate (supply an user and password) to the server at the beginning of the session. Again, the problem is the lack of good client support for this undocumented extension and IMAP (Internet Message Access Protocol) does not currently have a similar feature.

Using a combination of the ideas above it is possible to require the user to authenticate to the server before they are allowed to relay email using a “POP(IMAP)-before-SMTP” policy without additional client support. POP(IMAP)-before-SMTP relaying works like this: every time someone successfully enters a correct username and password to the POP/IMAP server, the POP/IMAP server records the IP address of remote client for later use by the SMTP server. This IP address is stored in a .db file (`etc/relayers.db`) with a timestamp of the login. This database will serve as a list of IP addresses that are allowed to perform an SMTP relay and is used by sendmail during an SMTP transaction. Placing a simple set of rules in the “`check_rcpt`” section of the `sendmail.cf` file will cause sendmail to refuse to relay except for IP addresses recorded by the POP/IMAP daemon.

It is important to note that POP(IMAP)-before-SMTP method requires the user to use POP/IMAP (check their mailbox) first before they send any email. For example, here is a sample SMTP relay transcript showing the initial refusal by sendmail to relay:

```
% telnet smtp.yourcompany.com smtp
Trying 192.168.1.100...
Connected to smtp.yourcompany.com.
Escape character is '^]'.
220 smtp.yourcompany.com ESMTP Sendmail 8.8.5 ready at Thu, 29 Jan 1998 12:26:04
HELO spammer.com
250 smtp.yourcompany.com Hello smtp.spammer.com [192.168.1.10], pleased to meet you
MAIL FROM:<foofoo@spammer.com>
250 <foofoo@spammer.com>... Sender ok
RCPT TO:<someone@somewhere.com>
550 <someone@somewhere.com>... SMTP relay denied, authenticate via POP first
QUIT
221 smtp.yourcompany.com closing connection
Connection closed by foreign host.
```

Here is a sample SMTP relay transcript showing a successful POP login and email relay:

```
% telnet pop.yourcompany.com pop
Trying 192.168.1.10...
Connected to .
Escape character is '^]'.
+OK QPOP (version 2.2) at pop.yourcompany.com starting.<22485@pop.yourcompany.com>
USER foofoo
+OK Password required for foofoo.
PASS *****
+OK foofoo has 0 messages (0 octets).
QUIT
```

```

+OK Pop server at pop.yourcompany.com signing off.
Connection closed by foreign host.

% telnet smtp.yourcompany.com smtp
Trying 192.168.1.10...
Connected to smtp.yourcompany.com.
Escape character is '^]'.
220 smtp.yourcompany.com ESMTP Sendmail 8.8.5 ready at Thu, 29 Jan 1998 12:27:09
HELO somewhere.com
250 smtp.yourcompany.com Hello smtp.somewhere.com [192.168.100], pleased to meet you
MAIL FROM:<foofoo@yourcompany.com>
250 <foofoo@yourcompany.com>... Sender ok
RCPT TO:<someone@somewhere.com>
250 <someone@somewhere.com>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
From: Foobar Foo <foofoo@yourcompany.com>
To: Someone <someone@somewhere.com>
Subject: SMTP relay

This is an SMTP relay!
.
250 MAA22708 Message accepted for delivery
QUIT
221 yourcompany.com closing connection
Connection closed by foreign host.

```

Blue Reef's Virtual Server includes support for creating the etc/relayers.db file built-in to the POP and IMAP servers.¹ It does require the additional check_rule in your sendmail.cf if you don't already have it:

```

Krelayers hash /etc/relayers.db
Kdequote dequote

#####
# Anti-Spam Support: Limit SMTP relaying to previously authenticated users #
#####
Scheck_rcpt
R< $+ @ $=w >[tab] @$ OK
R$+[tab]          $: $(dequote "" ${client_addr} $) $| $1
R0 $| $*[tab]     @$ OK
R$* $| $*[tab]    $: $(relayers $1 $: ERROR $)
RERROR[tab] $#error $@ 5.7.1 $: "550 SMTP relay denied,authenticate via POP/IMAP 1st"
R$*[tab]          @$ OK

```

Again, the [tab]'s should be replaced with the tab character. Here sendmail allows SMTP recipients that are identified as local users and rejects all other messages unless the sendmail was invoked locally or the sender's IP address is in the relayers.db file.

While the POP and IMAP servers automatically add to the etc/relayers.db file you may on occasion want to manually add or clean the database yourself. To do this Blue Reef has a custom-written utility named "vsmtprelay" that can add, delete, expire, or list IPs in the etc/relayers.db file. Here is the help screen:

```

% vsmtprelay

vsmtprelay 1.0.0 usage (optional items in []):

    % vsmtprelay command [arg] [...]

where "command [arg] [...]" can be one of the following:

```

1. Support for the POP/IMAP-before-SMTP feature is included with servers ordered after March 1, 1998. Most all of the older virtual servers were upgraded to add this support as well. If you think your server does not have this feature installed please contact support.

```

"add ip [ticks]"   insert address with current timestamp (or ticks)
"delete ip [...]"  remove specified address(es)
"expire [n]"       expire all (or older than n minutes) SMTP relayers
"list [n]"         list all (or older than n minutes) SMTP relayers

```

IP addresses are expressed as ASCII "dotted quads", e.g. "192.41.1.10". All timestamps are stored as ASCII strings representing a count of seconds elapsed since 0 hours, 0 minutes, 0 seconds, January 1, 1970, Coordinated Universal Time (the common UNIX epoch).

Note that the relayers.db file is implemented as a "Berkeley DB hash file" with IP addresses as the key and the timestamp as the data (all as ASCII strings). Here is sample output from the "list" command:

```

% vsmtprelay list
# timestamp (UTC): Mon Jan 26 19:42:10 1998
192.168.1.10      885843730

# timestamp (UTC): Mon Jan 26 19:43:40 1998
10.100.220.100   885843820

# timestamp (UTC): Mon Jan 26 19:47:26 1998
192.168.1.220    885844046

# timestamp (UTC): Tue Jan 27 19:56:05 1998
192.168.1.124    885930965

# timestamp (UTC): Mon Feb  2 18:22:37 1998
10.128.55.124    886443757

```

The output is intentionally produced in a form that can be edited manually and rebuilt by makemap(8) if desired. Here are other usage examples:

```

# add 192.41.1.10 with a timestamp of Wed Feb  4 10:53:40 MST 1998
% vsmtprelay add 192.41.1.10 886614820

# add 192.41.1.10 with a current timestamp
% vsmtprelay add 192.41.1.10

# delete 192.41.1.10 from the database
% vsmtprelay delete 192.41.1.10

# delete 192.41.1.10 and 206.107.170.2 from the database
% vsmtprelay delete 192.41.1.10 206.107.170.2

# expire all addresses from the database
% vsmtprelay expire

# expire all addresses older than 10 minutes from the database
% vsmtprelay expire 10

# list all addresses in the database
% vsmtprelay list

# list all addresses older than 10 minutes in the database
% vsmtprelay list 10

```

Even though the database does not take a lot of space on the virtual server you may wish to automatically expire entries on a periodic basis to keep the database small. This can be done with cron (see "man crontab"). For example, the following cron entry would expire all the entries that are a day old every night at 3:15am:

```

15 3 * * * /usr/local/bin/vsmtprelay expire 1440

```

4.0 Summary

While commercialization of the Internet has brought many benefits, among its negative effects is the proliferation of unsolicited commercial email. While UCE was once a rare item, it has since grown to the point that many active Internet users are targeted by spammers on a daily basis, often with offensive or obnoxious commercial offers. UCE, like "junk faxes", should eventually be outlawed in the United States and in other parts of the world. Until then, the methods in this paper can be used to effectively block spammers from abusing your Virtual Server.

References:

- [1] <http://spam.abuse.net> ("Fight Spam on the Internet!")
- [2] <http://www.cauce.org> ("The Coalition Against Unsolicited Commercial Email")
- [3] <http://www.arachnoid.com/lutus/antispam.html> (P. Lutus' Anti-Spam page)
- [4] <http://www.sendmail.org/antispam.html> (sendmail anti-spam ideas)
- [5] <http://www.best.com/~ariel/nospam/> (procmail anti-spam ideas)
- [6] <http://spam.abuse.net/tools/smPbS.html> ("POP before SMTP for Sendmail")
- [7] <http://www.informatik.uni-kiel.de/~ca/email/check.html> ("Using check_* in sendmail 8.8")
- [8] "sendmail, 2nd Edition" by Bryan Costales & Eric Allman, O'Reilly & Associates (See <http://www.ora.com/catalog/sendmail2/noframes.html> for more information)
- [9] "Master Regular Expressions" by Jeffrey E.F. Friedl, O'Reilly & Associates (See <http://www.ora.com/catalog/regex/noframes.html> for more information)
- [10] "DNS and BIND, 2nd Edition" by Paul Albitz & Cricket Liu, O'Reilly & Associates (See <http://www.ora.com/catalog/dns2/noframes.html> for more information)
- [11] <ftp://ds.internic.net/internet-drafts/draft-myers-smtp-auth-09.txt> and <ftp://ds.internic.net/internet-drafts/draft-gellens-submit-05.txt> Internet drafts on extending SMTP to include authentication.
- [12] <http://web.syr.edu/~jmwobus/comfaqs/lan-mail-protocols.html> "Serving Desktop Computers Using a Central Mail Server on an Internet" by John Wobus