;login:

THE MAGAZINE OF USENIX & SAGE

February 2001 • volume 26 • number 1

inside:

ROBERT HASKINS
ISP ADMIN:
MAIL ARCHITECTURE

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

ISPadmin

Mail Architecture

by Robert Haskins

Robert Haskins is currently employed by WorldNET, an ISP based in Norwood, MA. After many years of saying he wouldn't work for a telephone company, he is now affiliated with one.

<rhaskins@usenix.org>

INTRODUCTION

In this column, I will cover various topics that are in some way unique to the Service Provider (SP) industry. Before working in the ISP industry, I often wondered how SPs handled problems like high-volume mail or news, Web hosting, etc. I will attempt to illustrate how many SPs engineer various services for this often high-volume, high-expectation industry. The following topics may be covered (in no particular order) in future columns:

- RADIUS
- LDAP
- Provisioning/billing
- DNS
- News
- Security
- Web caching
- Web hosting
- Network monitoring/SLAs

I will use the various Service Providers I have worked for in the past as the primary case studies, including Time Warner Cable of Maine and Ziplink, Inc. I will also attempt to cover alternate case studies as well, where appropriate.

The Problem of Mail at a Service Provider

In this installment, I will look at how mail solutions are architected. At any SP, implementing a robust mail architecture is different from a typical enterprise for the following reasons:

- High volume of mail
- Many customers utilizing mail
- High expectations, as this is sometimes a pay-for service

Now, that is not to say that some enterprise mail systems can't have the above characteristics; they certainly can. It's just that these characteristics define any SP's mail architecture.

I would be willing to bet that the reason most people obtain Internet access is first and foremost to read and send email. Sure, they want to surf the Web, but ask most subscribers what's the most important application they use when online and I'm sure they'd answer "email." This popularity translates into lots of email going to and from many subscribers. The proliferation of email-based greeting cards, jokes, hoaxes, spam, etc., only serves to put additional pressure on SP's mail infrastructure. Let's start by examining how an enterprise might engineer their mail system.

A Simple Example

A small- to medium-sized enterprise has different goals than an ISP when it comes to designing a mail infrastructure. However, it is still worthwhile to compare how most other enterprises' mail setup compares to an SP mail infrastructure. I will assume that this imaginary enterprise is behind a firewall for security purposes. Their mail system might be set up like the diagram in Figure 1.

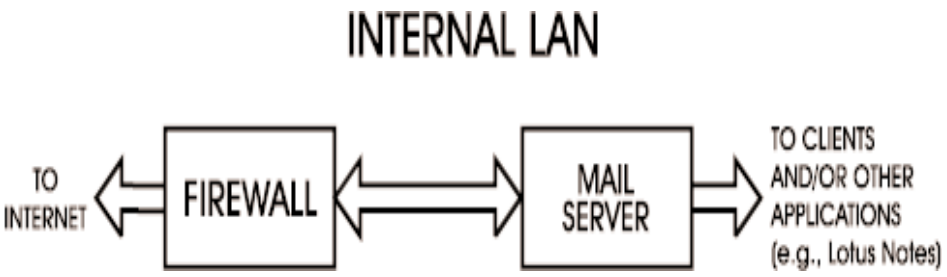

Figure 1

In most of the enterprises I am familiar with, the firewall only accepts outbound mail connections from the internal mail server on the secure interface to limit exposure to potential security problems. However, one could easily set up the firewall to accept outbound connections from any internal client originating on the secure interface. The firewall must always accept inbound mail from anyone (except perhaps those servers listed in the Mail Abuse Prevention System's [MAPS] or similar anti-spam "black hole" lists if the site chooses to subscribe to such a service) coming in on the insecure interface on port 25. In any case, the firewall must function as inbound and outbound mail relays would work in an SP environment, while the single mail server machine handles all other mail functionality. This single mail server machine ends up being a major bottleneck in an SP environment. To address this shortcoming, the problem of mail is decomposed into its smaller pieces, which is the topic of the next section.

Breaking Down the Problem of Inbound Mail

The way mail is engineered at SPs is to decompose the process into smaller, scalable pieces. Mail functionality can be broken down into these categories:

- Relaying
- Storing/end user retrieval of messages
- Forwarding mail
- Mailing lists
- Bouncing mail for unknown users

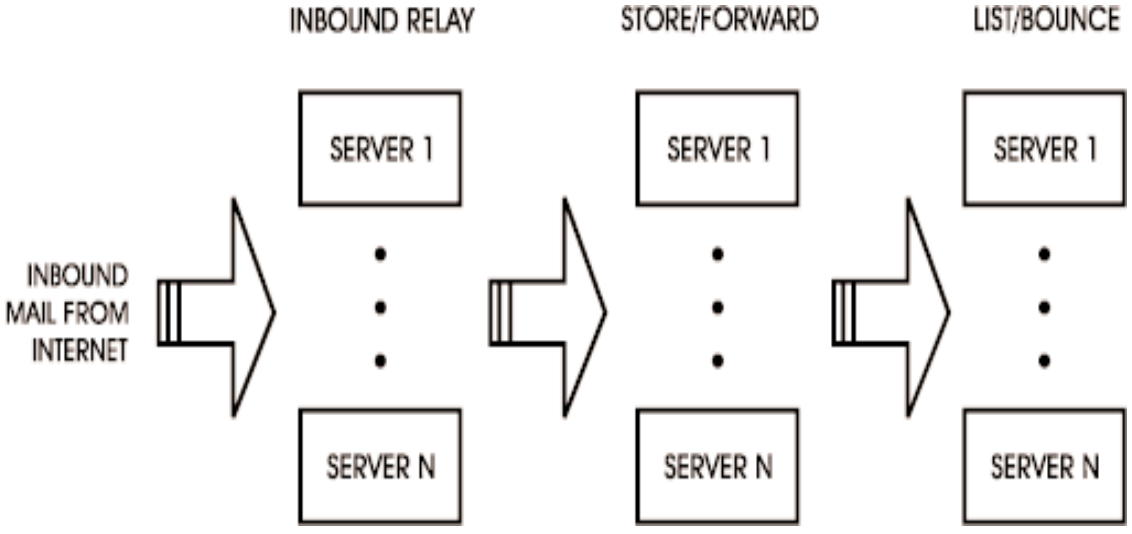

Figure 2

Figure 2 demonstrates how a relatively large ISP might engineer an inbound mail solution. It requires a bit of explanation prior to going into detail on each particular part. The arrows in Figure 2 illustrate the flow of inbound mail messages. The ellipses indicate that the functionality is scaled depending upon the load; for example, there is no need to have the same number of relay machines as store/forward machines. Each function is scaled depending upon the requirements of that particular service. Mailing-list maintenance and bounce functionality loading is relatively light and, as a result, would most likely be the last machine functions to require scaling. It is important to note that within a particular class of machine (relay, for example), the servers are essentially clones of one another, and can be brought up and down at will (ensuring appropriate queues get processed, of course). The message store is usually designed to access a shared file system (NFS, SAN, etc.) for the messages. This system is engineered with an appropriate level of redundancy within the file system in order to alleviate any possible single point of failure.

Most very large ISPs split inbound and outbound mail relays and have multiple machines dedicated to each type of functionality spread across their network.

INBOUND MAIL RELAYING

Most ISPs have one or more machines dedicated to mail relaying. In fact, most very large ISPs split inbound and outbound mail relays and have multiple machines dedicated to each type of functionality spread across their network. In this context, inbound mail refers to mail coming from other places (i.e., Internet or other WAN) destined for an end customer of that ISP. Outbound refers to mail originating on the ISP's network destined for another network.

In Figure 2, mail from the Internet at large would hit a series of dedicated inbound mail relays. These inbound mail relays might perform some sort of basic anti-spam checking (for example, check for the originating network to be listed in Mail Abuse Prevention Project's MAPS' Real time Black hole List, a.k.a. MAPS RBL, or the relays might run Blackmail software for domain and other message/header validation). Once these basic checks are performed, the mail is forwarded.

Typically the server software for relay functionality is Sendmail, although other mail server software can be, and is, used. The setup of such inbound mail relays is relatively straightforward, as it is a relatively simple problem to send mail from point "A" to point "B." The mail relay servers would need to know what domains it is accepting mail for (these would be hosted domains, of course) and forward the message to the appropriate mailbox. Typically, this is done through a UNIX db file and Sendmail setup. However, with the advent of directories, LDAP is a much easier and scalable way of solving what domain mail goes where.

STORE/FORWARD (AND A WORD ABOUT PROVISIONING)

The mail relays would then pass messages to a series of store/forward machines, which accept and deliver mail locally for legitimate users and forward mail for customers who choose to retrieve their mail from some other server. This is a relatively easy problem to solve for a small network. However, when the number of mail accounts exceeds several thousand or so users, the directory lookups can take so much time that an alternate scheme for storing messages must be deployed. The discussion here is centered upon a POP3 solution; the topic of IMAP will not be addressed.

In the past, the method used to address scaling of services as it pertains to mail storage was to exploit POP3 proxy functionality and forward the request to the appropriate machine by using some sort of a database updated by the provisioning process. I must

digress here and explain a little about what provisioning is. Provisioning is simply setting up subscriber accounts. It usually means performing the following steps:

- Creating a UNIX account with an invalid shell on a mail machine for mail retrieval by customer
- Setting up a UNIX account on an FTP server so a customer can update his/her Web site
- Configuring an Apache Web server home directory for the customer
- Etc. . . .

A full discussion of provisioning is out of the scope of this article. I will speak to this topic in a future column.

Besides utilizing the POP3 proxy functionality mentioned above, a more recent development in the area of scalability would be to use LDAP to determine exactly what machine the customer's mail resides on. The advent of the Pluggable Authentication Module, or PAM, makes utilizing LDAP a much easier proposition than it was before PAM arrived on the scene. Once again, a full discussion of PAM and LDAP deserves its own column and is beyond the scope of this discussion. The references section contains some links to resources on integrating Sendmail with LDAP and PAM.

A typical mail store would run Sendmail to receive mail and Qpopper to allow POP3 access by end subscribers. These machines need to be controlled by the provisioning process so they know which subscribers are active and which to bounce. They would also utilize some sort of a shared file system (SAN, NFS, etc.) so that the load on the message stores can be scaled easily.

MAILING LISTS/BOUNCING MAIL

The final step would be to have the mail-store machines forward mail destined for unknown recipients to a machine or set of machines dedicated to list processing, and to bounce any message that wasn't addressed to a hosted list. Typically, this is a machine running vanilla Sendmail and Majordomo list processing software. If the message is a hosted list, the list is expanded and sent to the mail store and outbound mail relays for final delivery. If the message is not a hosted list, then the message is bounced back to the sender, since it is undeliverable.

Typically, this functionality doesn't take a lot of resources, so this would be the last machine to require scaling. Also, it is relatively straightforward to configure. It does not require access to the provisioning process, and can easily scale without a need for a shared file store or other such complications.

OUTBOUND MAIL RELAYING

Outbound mail refers to clients sending mail to the outside world. Inbound and outbound mail relays can be the same machine. The only additional functionality performed by an outbound mail relay is an address range check to ensure that only end subscribers of the SP can relay mail through the machine. If this check were not made, any arbitrary user could send mail through the relay, which is known as an "open relay" and is a "Very Bad Thing."

MAIL SERVER SOFTWARE BESIDES SENDMAIL

As I have previously mentioned, most SP installations utilize Sendmail. I think the reason for this is a testament to how robust and flexible Sendmail has proven over the

years. However, there are other solutions out there, in use by SPs. Freeware mail server software would include:

- Qmail
- Postfix
- Exim

Commercial solutions include:

- Intermail Post.Office from Openwave Systems, Inc. (formerly software.com)
- PMDF from Sun/Netscape Alliance (formerly Innosoft, Inc., now supported/developed by Process Software, Inc.)
- CommuniGate Pro from Stalker Software, Inc.

While I have no direct experience with any of the above solutions (either freeware or commercial), I am certain they all can be made to work in SP environments.

SPAM

No discussion of SP mail solutions would be complete without including the topic of spam. The problem of spam can be broken down into two parts: inbound and outbound. Most if not all available solutions today address the problem of inbound spam; I am aware of no commercially available solution that tackles the specific problem of outbound spam.

There is some anti-spam support within recent versions of Sendmail. Here is a list of some of the features within Sendmail 8.10:

- Anti-spam rule sets
- Content-based filtering
- Built-in SMTP authentication
- RFC2505 support
- RFC2476 (Mail Submission Agent specification)
- Specific senders/recipients can be allowed or disallowed Sender/recipient-based filtering

However, the Sendmail anti-spam functionality does not go far enough for most ISPs, so additional pieces must be added. Some available third-party freeware available includes:

- Blackmail (implements many of the recommendations in RFC2505)
- Spamshield (counts log file entries for users sending large amounts of mail and can stop them in real time if desired)

Another methodology for blocking spam is to utilize a service such as Brightmail. The Brightmail Logistical Operations Center has spam forwarded to it from "mail probes" located at SPs around the world. Their staff generates rule sets for their spam-blocking software that works in conjunction with an ISP's mail infrastructure. These rule sets identify specific pieces of "Unsolicited Commercial Email" and "sideline" them for later perusal by the end subscriber. This service can be a very effective method of blocking inbound spam. Note that Brightmail also offers a free service which blocks mail via POP3 proxy. You can find more information under the "Brightmail Individual" heading on the Brightmail Web site.

I am aware of no commercially available solution that tackles the specific problem of outbound spam.

REFERENCES

Mail Abuse Prevention System: <*http://www.mail-abuse.org*>

Sendmail.net (articles on using Sendmail): <*http://www.sendmail.net*>

Sendmail Consortium (freeware): <*http://www.sendmail.org*>

Blackmail: <*http://bitgate.com/spam*>

LDAP man (articles on configuring LDAP): <*http://www.ldapman.org*>

Linux-PAM: *http://www.lyre-mit-edu.lkams.kernel.org/pub/linux/libs/pam/*

Qpopper: <*http://www.eudora.com/qpopper/*>

Majordomo: <*http://www.greatcircle.com/majordomo/*>

Qmail: <*http://www.qmail.org/*>

Postfix: <*http://www.postfix.org/*>

Exim: <*http://www.exim.org*>

Intermail Post.Office: <*http://www.openwave.com/index.html*>

PMDF: <*http://www.innosoft.com/*>

CommuniGate Pro: <*http://www.stalker.com/*>

IETF RFC tool: <*http://www.ietf.org/rfc.html*>

Blackmail: <*http://bitgate.com/spam*>

Spamshield: <*http://spamshield.conti.nu*>

Brightmail: <*http://www.brightmail.com/*>

Dealing with Large Amounts of Mail

One of the problems faced by both enterprise and SP system administrators alike is how to deal with large volumes of mail. In an SP environment, the abuse mailbox can easily run into the thousands of messages per day. This doesn't include the mail that system accounts such as "root" generate on a typical day. (Of course, ISPs typically have a Network Operations Center or other support personnel who (are supposed to) respond to abuse complaints in a timely fashion.) I have used two methodologies when dealing with system (non-abuse) mail, neither with much success:

1. Forward all mail to a central location and read it from there.
2. Read all mail locally.

The issue with 1 is that under certain conditions, the volume of messages can easily bring down even the most robust mail system. The issue with 2 is how to read system mail on 200 servers each day and get some productive work accomplished. If anyone has any thoughts on methods to deal with this topic, I'd love to hear from you.

CONCLUSION

A Service Provider's mail infrastructure must be designed for robustness and scalability. Robustness is handled by utilizing time-proven hardware, software, and designs. Scalability is achieved by decomposing the problem of handling mail down into its component problems: relay, storage, bounce, etc.

Next time I'll cover the little known topic (outside of the ISP industry) of Remote Authentication Dial-In User Services, or RADIUS. In the meantime, please send your questions or comments on this column, UNIX systems administration, or any other related topic to me! I'd love to hear from you.