

Just Can't Get Enough Sendmail

Hal Pomeranz

In the June issue of *Sys Admin*, I discussed the concept of shutting down the standard Sendmail daemon on most Unix machines. To recap, this daemon is responsible for two things:

1. Listening on port 25/tcp for *incoming* messages from outside of the machine
2. Flushing the local queue of unsent messages on a periodic basis

If the machine is not a mail server, then there is no need for your system to be listening on port 25/tcp for incoming email (flushing the queue of outgoing messages can be handled by running Sendmail from cron on a periodic basis). Disabling the Sendmail daemon on most of your machines is a huge security win, since it prevents external attackers from using Sendmail as a mechanism for breaking into your site.

This article resulted in an enormous amount of feedback, but there were a small number of specific issues that seemed to come up over and over again. Clearly there are many sites that are grappling with similar issues. After some discussion with the editors, we thought that it might be a good idea to write a follow-up article that covered some of these common issues in more detail.

Black-Holes, Bounced Email, and Replies

After implementing my suggested configurations on their systems, several people noticed that they were no longer receiving email that got sent to root or other users by the cron daemon or any other processes running on the local machine. Other people noted that mail bounces and error messages, and even normal replies to messages sent from the reconfigured machine were not making it back to the original sender. What's happening here is "normal behavior" for Sendmail, but obviously not desirable behavior for the affected sites.

Given a "bare" address like root or mary, Sendmail will automatically add the fully qualified hostname of the local machine on the righthand side of the address. So mail to root will end up being sent to root@somehost.yourdomain.com. Now, per the configuration guidance in my previous article, this email will get forwarded to some central relay server elsewhere on your network for final delivery. Unfortunately, this relay server is going to attempt to send the email right back to the machine that generated it, because the delivery address is the machine somehost.yourdomain.com. But that machine is no longer running a Sendmail daemon to receive this incoming email! So the message sits in the outgoing mail queue on the relay server until it times out and is expunged. The mail never gets delivered.

Note that Sendmail also qualifies the sender address in the same way. So outgoing messages leave the machine as being from user@somehost.yourdomain.com. If somebody tries to reply to this message, the response will end up sitting in a mail queue someplace trying to be delivered to somehost.yourdomain.com again, and ultimately never be delivered. The same thing happens to bounce messages and other errors.

This is clearly not what we want. Most sites deal with this problem by "hiding" their host names, or *masquerading* in Sendmail lingo. In other words, bare addresses get qualified with just yourdomain.com, rather than the full hostname of the machine where the email message was generated. Now all messages, replies, and bounces will simply go to user@yourdomain.com, and this traffic can easily be handled by your central mail servers.

Just Can't Get Enough Sendmail

Hal Pomeranz

In order to fully enable masquerading, simply add a few following configuration directives to your Sendmail macro configuration file. These directives can be included in either the nullclient.mc or submit.mc file discussed in the previous article:

```
MASQUERADE_AS(`yourdomain.com')  
FEATURE(`allmasquerade')  
FEATURE(`masquerade_envelope')  
FEATURE(`always_add_domain')
```

The MASQUERADE_AS directive specifies the domain that should be used to qualify bare addresses instead of the fully-qualified hostname of the local machine. Usually this is your local domain name, but it could be any domain name you wish to specify, even a domain that is foreign to the local site. Now by default Sendmail masquerading only applies to the outgoing sender address. The allmasquerade feature tells Sendmail to use the masqueraded domain on recipient addresses as well. Furthermore, it's necessary to use masquerade_envelope so that the "envelope sender" address is also masqueraded, because this address is where bounces and replies will be sent.

The always_add_domain feature is not strictly related to masquerading. This directive tells Sendmail to always qualify both bare recipient addresses (e.g., mail sent to root from the cron daemon) as well as the outgoing envelope sender address. If MASQUERADE_AS has been used to specify a domain name then that name will be used, otherwise Sendmail just uses the fully-qualified hostname of the machine. Qualifying all addresses in outgoing mail (whether you're using masquerading or not) is generally thought to be "good practice".

The Old "Bind to Loopback" Trick

A number of the responses I received to the original article described a configuration where the site leaves the Sendmail daemon running on their machines, but instead of having it listen on port 25/tcp on all of the network interfaces on the system, they configure the daemon to only listen to 25/tcp on the internal software "loopback" interface of the machine. This is the interface with IP address 127.0.0.1 on your system, which is only accessible to processes running on the local machine. Several of the emails I received noted that this is the default configuration for RedHat and possibly other Linux distributions.

This configuration certainly accomplishes our primary security goal, because the Sendmail daemon is no longer available to an external attacker. On the other hand, this configuration still allows a local user-- or an attacker who has gotten access to the local machine as an unprivileged user-- to use the local Sendmail daemon as a potential privilege escalation path to "break root" on the system. Perhaps this is not a huge issue in the home/hobby environment, but it could be a significant problem for large enterprises and high-security environments.

The "bind to loopback" configuration does make things easier for sites upgrading to Sendmail v8.12, since the default Message Submission Process (MSP) wants to deliver outgoing email to a Mail Transfer Agent (MTA) listening on 25/tcp at the loopback interface. Perhaps this is why RedHat chose this as their default configuration.

The way to configure Sendmail to listen on a specific address and port number is with the DaemonPortOptions in the sendmail.cf file:

Just Can't Get Enough Sendmail

Hal Pomeranz

```
# SMTP daemon options
o DaemonPortOptions=Addr=127.0.0.1,Port=smtp,Name=MTA
```

The configuration line you see here forces Sendmail to listen on the smtp port (usually 25/tcp as defined in /etc/services) on the loopback interface (address 127.0.0.1).

If you prefer, you may also set this option in your m4 macro configuration file. If you are using Sendmail v8.11 or later, then use the following configuration directive:

```
DAEMON_OPTIONS(`Addr=127.0.0.1,Port=smtp,Name=MTA')
```

For versions prior to v8.11, you use:

```
define(`confDAEMON_OPTIONS',`Addr=127.0.0.1,Port=smtp,Name=MTA')
```

In either case, the sendmail.cf file you generate should have DaemonPortOptions set appropriately.

Local Alias Expansion

A number of people wrote in to say that my configuration suggestions had "broken" aliases they had set up on their local systems. For example, several sites had a local alias set up for root so that this email would end up going to the specific admin for that machine. Once the system was reconfigured per my instructions, the alias was no longer being resolved and email was going unexpected places. Again, this is "expected" behavior, though maybe not what you wanted. Alias expansion is handled by Sendmail's "local delivery" mailer. However, the configuration in the previous article turned the local machine into a simple mail relay, completely bypassing any attempt at local delivery on the system. This means that alias expansion will never happen.

I would argue strongly that relying on specific aliases configured on individual client systems scattered throughout your organization is poor configuration practice. You are administratively much better off centralizing your aliases on your primary mail servers wherever possible.

Still, perhaps you don't control the central mail servers or have some other good reason for needing to use the local aliases file. In this case, you must run a mail server on the local machine to handle "local delivery"-- at least as far as expanding an address from the local aliases file and then delivering the email to its final destination on some other host. This is probably another good use for the "bind to loopback" configuration described in the previous section, since there's usually no reason for the system to ever accept incoming email from outside the machine.

Conclusions

Masquerading is clearly a useful addition to the concepts I covered in my original article. Note that you will typically use masquerading on your mail servers as well, since you would like hostnames to be hidden on all email originating at your site.

I admit to being highly ambivalent about running the Sendmail daemon bound to the loopback interface only. This configuration may make sense in SOHO situation where you might not have an external mail server that you can use as a relay, or in the case where you need to do local alias expansion on the machine. In an enterprise type environment, however, I vastly prefer to simplify things by disabling the Sendmail daemon on all but my central mail servers.