

# Improving Sendmail Security by Turning It Off

Hal Pomeranz

Sendmail is, for better or worse, the de facto standard Mail Transfer Agent (MTA) for UNIX systems. While many books could be written about the pros and cons of replacing Sendmail with an alternate MTA -- such as Postfix, Qmail, or Exim -- the reality is that most UNIX shops have a huge installed base of machines running Sendmail daemons as part of their default install. The surprising news, however, is that the vast majority of these systems do not need to be running a Sendmail daemon at all.

This issue is likely to become a critical one for sites in the near future. The Sendmail buffer overflow exploit announced in March will almost certainly be programmed into an automated worm within the next six months. Such a worm could do for UNIX systems what Code Red did to the Windows world, simply because there are so many potentially vulnerable UNIX systems on the network today.

Shutting off the Sendmail daemon on 99.9% of the systems in your environment would greatly reduce the potential impact of such a worm.

## The Role of the Sendmail Daemon

When I discuss security issues with systems administrators, I find that many of them are confused about the need for a running Sendmail daemon on their systems. This confusion is understandable since, for at least the last two decades, the commercial UNIX vendors have been shipping their operating systems with active Sendmail daemons in the default install. Most administrators simply assume that this daemon is necessary for the users and automated processes running on the system to be able to emit email from the machine.

The reality, however, is that the Sendmail daemon on a machine is only 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

Having a Sendmail daemon listening on 25/tcp opens the system up for remote exploits such as the recently announced buffer overflow issue. Note that the system only needs to be listening on 25/tcp if it is actively expecting to receive email messages from other systems. In fact, the only systems on the network that receive external email are the machines that are acting as mail servers and mail relays. A given site will typically only have a small handful of these machines (many of which are Windows machines running Exchange anyway). The other 99.9% of the systems on the network are email "clients" -- that is they may emit email messages from time to time, but never expect to receive a message from another machine. On these machines, it is probably best to simply disable the Sendmail daemon altogether in order to eliminate the potential for remote exploits.

But what about outgoing email generated from the local system? UNIX email clients send outgoing email by invoking the Sendmail binary directly off the disk. This new Sendmail process simply attempts to deliver the message to its next hop directly, rather than communicating with the local Sendmail daemon running on the system. So, while it is necessary to have a proper Sendmail configuration file on each local system, it is not necessary to have a Sendmail daemon running to allow for users and processes on the system to send email out of the machine. The release of Sendmail v8.12, however, has complicated this simple view of the world somewhat, as I will discuss later in this article.

# Improving Sendmail Security by Turning It Off

## Hal Pomeranz

Remember that the local Sendmail daemon is also responsible for managing the local queue of unsent messages. For example, when a user creates an email message, the local Sendmail process invoked by the user's mail client may not be able to reach the next hop mail server immediately. This message is then queued in the system's local mail queue. If the local Sendmail daemon on the system is shut off, then there is nothing to process the local mail queue, and this unsent message will never be delivered! So when shutting off the local Sendmail daemon to avoid remote exploits, it is important to arrange for some mechanism to flush the local mail queue on a regular basis.

So let's tackle all of these issues. To begin, I'll look at how to manage the local mail queue and then discuss appropriate Sendmail configurations for "client" type machines -- both for older versions of Sendmail and the new v8.12 release.

### Dealing with the Queue

The standard invocation for Sendmail (assuming you're running a Sendmail version prior to v8.12) typically looks something like this:

```
/usr/sbin/sendmail -bd -q15m
```

The **-bd** flag is what tells the Sendmail daemon to listen on 25/tcp for incoming email. The **-q15m** flag tells the Sendmail daemon to attempt re-delivery of queued messages every 15 minutes.

One option is simply to run the Sendmail daemon without the **-bd** flag. The daemon will still run and flush the queue every 15 minutes, but it will not actively listen on 25/tcp for incoming messages. Newer UNIX operating systems make it easier for administrators to configure their systems to run in this mode. For example, starting with Solaris 8, the **/etc/default/sendmail** file can be used to configure this behavior (set **MODE=""**), and Red Hat has implemented a similar mechanism in **/etc/sysconfig/sendmail** (set **DAEMON=no**). If your operating system vendor supplies a mechanism for configuring this behavior, it's probably best to go with the vendor-supported interface.

On the other hand, some UNIX systems present only the simple choice of having the daemon running and listening on 25/tcp or not running the daemon at all. Certainly the administrator could hack the appropriate boot script to remove the **-bd** flag from the Sendmail invocation, but it may be more maintainable in the long run to simply shut off the daemon altogether. But how do you make sure the queue gets flushed if the daemon is no longer running? One simple approach is to add a line to root's crontab:

```
0 * * * * /usr/sbin/sendmail -q
```

This will run **sendmail -q** at the top of every hour to attempt delivery of any unsent messages. This is probably a sufficient interval, since the queue should be empty under normal circumstances anyway.

### Local Sendmail Configuration

We still need to make sure that each system has an appropriate Sendmail configuration file, however, so that users and processes on the machine can send out email from the host. Generally, our email

## Improving Sendmail Security by Turning It Off

Hal Pomeranz

"client" type machines will need to simply relay all outgoing email to some local mail server for processing and routing, which is exactly what the standard "nullclient" configuration is designed to do:

```
include(`cf/m4/cf.m4')
define(`__OSTYPE__', `')
FEATURE(`nullclient', `mailhost')
```

The second argument to **FEATURE(nullclient)** is the name (or IP address) of the mail server to which outgoing email should be relayed. You will need to change this hostname to something appropriate for your site.

The other two lines in this configuration also deserve a little explanation. The **cf.m4** file contains all of the macro definitions, which expand items like **FEATURE(nullclient)** into the actual Sendmail configuration language used in the **sendmail.cf** file. The **cf.m4** file is found in the Sendmail source distribution under the **cf/m4** subdirectory; be sure to replace the pathname in the first line above with the correct pathname to the **cf.m4** file on your system.

The second line defines the **OSTYPE** macro to be the empty string. Normally, **OSTYPE** would be set to a string that indicates which OS platform the config file would be used on, and controls various OS-specific parameters (file locations, command-line arguments, etc.). However, for a simple nullclient configuration, none of that OS-specific information is important. So, we can just generate a completely generic configuration file that will work on any OS platform simply by setting **OSTYPE** to null.

To generate a Sendmail configuration file for your system from the above macros, simply type the above three lines into a text file, taking care to preserve the balanced left and right quotes and making the appropriate changes on the first and third lines for your local environment. Assuming the new file name is **nullclient.mc**, you can generate a **sendmail.cf** file with the following command:

```
m4 nullclient.mc > sendmail.cf
```

Install the resulting **sendmail.cf** file in the appropriate location on your system (usually **/etc/mail/sendmail.cf**, but sometimes just **/etc/sendmail.cf**).

### Dealing with Sendmail v8.12

Things have become more complicated as of Sendmail v8.12. This is because v8.12 is the first release of Sendmail to draw a distinction between the Message Submission Process (MSP) and the Mail Transfer Agent (MTA) portions of the email equation. Simply put, the MSP is the procedure used by local processes when they want to emit email from the local machine. The MTA is the process that handles receiving email from other systems and either relaying it onwards or delivering it locally. In other words, the MTA is the process that listens on 25/tcp and is the thing we want to shut off.

The problem is that the default behavior of the MSP is to attempt to deliver outgoing email by talking to the MTA over the system's internal loopback interface. Thus, if we shut off the local MTA, then the default MSP will be unable to send email out of the system. This is not a feature.

## Improving Sendmail Security by Turning It Off

Hal Pomeranz

It turns out, however, that the MSP doesn't have to emit email by talking to the local MTA over the loopback interface. It can talk to any mail server on the network that the administrator desires. In this mode, the MSP is acting very much like the nullclient configuration discussed in the previous section.

The macro configuration file is only slightly more complicated:

```
include(`cf/m4/cf.m4')
define(`confCF_VERSION', `Submit')
define(`__OSTYPE__', `')
define(`confTIME_ZONE', `USE_TZ')
define(`confDONT_INIT_GROUPS', `True')
FEATURE(`msp', `mailhost')
```

Again, set the pathname on the first line and the name of the machine on the last line as appropriate for your site. The configuration file produced from the above macros should be installed as **/etc/mail/submit.cf** on your systems.

Note that you can also simply hack the **submit.cf** file provided by your vendor. Just look for the line that reads:

```
D{MTAHost}[127.0.0.1]
```

Just replace **[127.0.0.1]** with the name of your new relay host. For example:

```
D{MTAHost}mailhost
```

Save your changes, and you're done.

The split between MTA and MSP is also reflected in how the Sendmail daemon is started at boot time. When using Sendmail v8.12, two daemons are normally started by the system boot scripts:

```
/usr/sbin/sendmail -bd -q15m
/usr/sbin/sendmail -Ac -q15m
```

The first line looks identical to the normal Sendmail invocation for Sendmail v8.11 and earlier -- this is the MTA process that is listening on 25/tcp for incoming email, and that processes a queue of messages received by this process. The second line invokes a daemon for the MSP. This daemon doesn't listen on any port for incoming messages: its sole duty is to process a separate MSP queue for messages generated on the local machine but that could not be delivered immediately because the MSP's mail relay was unavailable for some reason.

So in v8.12 Sendmail, the MSP has its own message queue and its own daemon for flushing that queue on a regular basis. This means that on email "client" type machines the MTA daemon is completely unnecessary and can be shut off completely. To get this to happen, however, it will probably be necessary to hack the default boot script provided by your vendor. I have yet to find a vendor who has a simple configuration switch for disabling the MTA process without disabling the

# Improving Sendmail Security by Turning It Off

Hal Pomeranz

MSP. In fact, most vendors in my experience don't even realize that you can run the MSP without a local MTA.

## Conclusion

Preventing the Sendmail daemon from listening on 25/tcp is an important security configuration step because it instantly protects you from remote Sendmail compromises. Please take this advice and disable your Sendmail daemons before an automated Sendmail worm is unleashed.

Of course, you will still need a process listening on 25/tcp on your mail server machines so they can receive and process incoming email, and you must be careful to stay up-to-date on Sendmail security fixes on these machines. Still, there's a huge difference between having to rapidly push out a Sendmail security fix to a handful of known critical email servers in your environment, and having to push that same patch to every single UNIX system in your environment just because they happen to be running Sendmail in daemon mode by default.