Global Information Assurance Certification Paper

Stopping Viruses at a Unix Mail Gateway

Thomas A. Heinrichs

August 20, 2001

Introduction

Many organizations run mail gateways on Unix, Linux, and *BSD. It is desirable to stop viruses at the mail gateway before they reach the recipients' mailboxes. To retain the trust of those we communicate with, it is also desirable to stop viruses in outgoing mail relayed out through the gateway. Although the Unix solutions aren't as widely available as those available for Exchange or Notes, it is possible to protect your users from viruses at a Unix mail gateway using both commercial and freely available tools.

The examples given in this paper refer to open source Sendmail running on Linux. However, many of the mail scanning agents also support other open source Mail Tranfer Agents (MTA), including Postfix, Qmail, and Exim. Regarding OS support for these tools, the open source options can generally be compiled under most *nix via autoconf support. However, the pre-compiled commercial tools that you will likely need to incorporate into your solution do not support as wide a range of operating systems. An unscientific survey indicates support is best for Linux followed by Solaris. It's hit-or-miss for other Unix and BSD OS's. Trend Micro appears to have the broadest Unix OS support.

Open Source versus Commercial Options

Administrators have various philosophical and economic reasons for choosing open source over commercial products. Fortunately, tools are available to implement a robust anti-virus mail gateway solution with open source tools for a cost from zero to tens of dollars per user.

However, even if you choose the open source route, you will likely need to incorporate, at a minimum, a commercial virus scanner into your solution. There are open source virus scanners available (SignatureDB), but they, by their own admission, will probably not live up to your expectations.

Administrators have various reasons for choosing commercial over open source products. The great hope for purchasers of commercial products is that, by paying for the product and "approved" extensions to the product (such as virus scanning), that the integration and support for the product will be superior to that of open source solutions. If you can get the right support person on the phone, a call to a vendor's tech support line to resolve a sticky problem can make the support contract cost money well spent.

Many open source products have commercial versions available—Sendmail, for example. It is possible to buy a shrink-wrapped copy of Sendmail and integrate it with a shrink-wrapped copy of Trend Micro's InterScan VirusWall for an out-of-box experience on

your HP-UX server. This may be the perfect solution for you, depending on your business environment and staff's skill and interests.

How the mail is intercepted for scanning

There are several points in the process of mail delivery that you can interpose virus scanning. In order to illustrate this, a simplified explanation of the mail delivery process goes like this:

1. A remote Mail Transfer Agent (MTA) such as Sendmail or Postfix contacts your mail gateway MTA via SMTP (port 25). The MTA's set up the connection, establish the existence of the user the mail is being sent to, then the remote MTA transfers the contents of the message to your mail gateway MTA.
2. Your mail gateway MTA looks at the address and decides what to do with the message. Depending on how the gateway is configured, one of two thing is likely to happen to the message:
    a. It is delivered locally to a mailbox (e.g., /var/spool/mail/john) for later retrieval by a Mail User Agent (MUA), such as Outlook Express, via POP or IMAP.
    b. Or, it is forwarded on to another MTA within your organization where the message will stored in a mailbox for user retrieval.

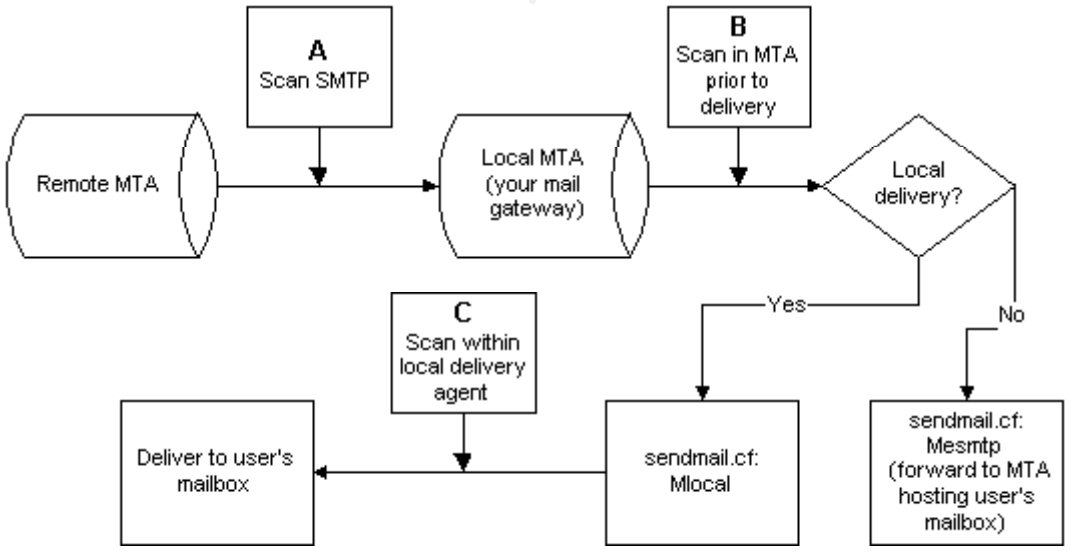

Figure 1. Simplified mail delivery process.

Intervention Point	Advantages	Disadvantages	Open source option available?
SMTP data stream (A)	- Can scan all mail, both incoming and outgoing. - Can be implemented with minor changes to existing MTA configuration. - Products can be used to scan other protocols: HTTP and FTP.	- Loads server. - Open source not currently available. - To scan outgoing mail, requires significant modifications to MTA configuration.	No
Within MTA prior to transfer to delivery agents (B)	- Can scan all mail, both incoming and outgoing.	- Loads server. - Requires significant modifications to MTA configuration.	Yes
By the local delivery agent (C)	- Scans all mail prior to delivery to users' mailboxes. - Scanning load can be distributed.	- Loads server. - Only incoming mail scanned.	Yes

Table 1. Comparison of e-mail scanning approaches.

So, where can virus scanning enter the process? First, the mail scanner can interpose itself prior to your MTA receiving the mail, scan it, then forward it on to your MTA for delivery (point A in figure 1). For example, this is the approach taken by Trend Micro's InterScan VirusWall product. It listens on port 25 for SMTP connections, receives the message, scans the message, then forwards the message on to the MTA (Sendmail), which is configured to listen on unprivileged, unused port, rather than the usual port 25. A disadvantage to this approach is that constant scanning of SMTP stream reduces server performance. Whether or not this is significant depends upon mail load and quality of service requirements. Use the vendor's "try before you buy" before committing.

Second, the mail scanner can interpose itself within your MTA prior to the MTA's decision about what will be done with the message (point B). This is useful for servers that are forwarding to the internal network because all incoming mail is scanned at its

entry into the network, regardless of its final destination. For example, Sendmail has implemented a mail filter library called "milter" that can be used for this purpose. The AMaViS scanning agent can use the milter interface in Sendmail to intervene and scan the message. A disadvantage is the performance hit on the gateway caused by scanning all messages. Also disadvantageous, you must make significant modifications to your MTA configuration. An additional, major processing step is inserted into the already rather complicated mail-processing stream within the MTA. It is unnerving to meddle with a well-functioning MTA.

Third, the mail scanner can interpose itself within your MTA at the local delivery agent level (point C). The mail gateway may be configured to only relay mail onto its final destination—departmental servers, for example. Each departmental server is running an MTA that delivers the mail locally to the end-users' mailboxes. The mail scanner can essentially enhance the local delivery agent so that, prior to deliver to the user's mailbox, it scans the message for viruses. For example, this is the approach taken by the "procmail-sanitizer". Procmail is a common local delivery agent for Unix with its own extensive programming language. Procmail-sanitizer uses a procmail "recipe" to look for potentially dangerous files and give them special treatment. The advantage to this is that the work of scanning can be distributed—for example, out to the departmental servers. The disadvantage is that this is only an incoming mail solution.

More detailed configuration information

SMTP data stream

These products are stand-alone. Some can also be configured to scan other protocols: HTTP and FTP, in particular. They can either be integrated into a firewall/proxy solution, or run stand-alone to handle a specific protocol.

To illustrate, Trend Micro's InterScan VirusWall runs as a daemon on port 25 and scans everything arriving via SMTP. After scanning the message, it then passes the message on to Sendmail (or any other MTA) listening on port 5000 (for example). This is accomplished in the configuration file, sendmail.cf, with the command:

```
O DaemonPortOptions=Port=5000
```

Interscan VirsusWall can also be configured to pass incoming mail received on port 25 directly to Sendmail using the "standard input" command line switch: sendmail –bs. This requires no changes to sendmail.cf.

Both of the above examples only handle incoming mail. They both also require minimal or no modifications to the Sendmail configuration. In order to accomplish scanning of both incoming and outgoing mail, the mailer definitions must be modified (Msmtp, Mlocal, etc.) to pass the all mail through the VirusWall scanner on alternate ports. See the vendor documentation for details.

Within MTA prior to transfer to delivery agents

To illustrate intervention within the MTA prior to delivery, two Sendmail examples are given. First, the Sendmail milter interface is specifically designed to filter incoming mail messages. The libmilter libraries come as part of the current Sendmail distribution. Using the libmilter API, products such as AMaViS, can create a filter accessible through the Sendmail option "InputMailFilters".

For example, the README.milter file in the AMaViS documentation shows the following changes to sendmail.cf :

> [Add] in the options section:
> O InputMailFilters=milter-amavis
> and in the mailers section at the bottom:
> Xmilter-amavis, S=local:/var/amavis/amavis-milter.sock, T=S:10m;R:10m;E:10m

This directs Sendmail to use the eXternal filter, milter-amavis, via a socket interface.

Milter has a number of well-thought-out design options, detailed in the "Architecture" section of the online documentation. These include performance enhancement (via multithreading), security (root not needed for filters), and robustness (default failover bypasses a faulty filter.) Currently, milter only implements filtering on incoming mail, although the developers leave open the possibility of filtering outgoing mail in future releases.

A second example is Kaspersky Labs' AVPKeeper utility. It is implemented via Sendmail rulesets and a custom mailer. Without going too far into the details, the Kaspersky modifications to the Sendmail configuration rewrite the destination address, appending .AVP to it. The .AVP extension is then routed to the "avpkeeper" mailer. The avpkeeper mailer performs virus scanning, then hands the message back to Sendmail. Sendmail then strips the .AVP extension to the address and sends the message on to the final destination. This approach depends upon the avpkeeper mailer to be robust. Messages that enter avpkeeper and, through a program fault, do not return to Sendmail are not delivered. All mail, both incoming and outgoing is scanned with this method.

By the local delivery agent

The local delivery agent can be a good place to filter messages in some situations. Most Unix anti-virus packages are fairly easily implemented here. Procmail-sanitizer is also implemented at this level. Although it is not a virus scanner, per se, it can offer a measure of protection to you users from viruses. A further benefit of procmail-sanitizer is that it is free.

The local delivery agent is used for delivery of mail to mailboxes that reside on the same system the MTA is running on. For example, it is specified by "Mlocal" in sendmail.cf

and “mailbox_command” in Postfix’s main.cf. Procmail and “mail” are two common local delivery agents.

As its name implies, Procmail-sanitizer works with procmail to screen incoming messages. Procmail has an extensive programming language of its own for developing procmail “recipes”. Procmail-sanitizer requires only that your local mailer be procmail, which is the default on most open source operating systems. Look in your sendmail.cf for something like:

```
Mlocal,      P=/usr/bin/procmail, F=lsDFMAw5:/|@qSPfhn9, S=10/30, R=20/40,
             T=DNS/RFC822/X-Unix,
             A=procmail -Y -a $h -d $u
```

Procmail-sanitizer is called through the global procmail run control file, /etc/procmailrc, every time procmail is invoked to make a local mail delivery.

Procmail-sanitizer is intended to block delivery of (in their jargon, “poison”) or render harmless (“mangle”) potentially harmful email attachments. Because Windows relies upon file extensions to signify a file is executable, modifying its extension can neutralize a potentially harmful file. For example, byebye.exe is changed to byebye-DEFANGED12345-EXE by Procmail-sanitizer. First, this prevents the user from simply double-clicking the message and causing harm. They are forced to first “Save As” the file to an executable extension. In this process, their local virus scanner has a better chance to examine the file. Second, any malicious script code embedded in the message body will not find the attachment by name nor be able to execute it.

AMaViS can also be configured to work at the local delivery agent level. In this mode, with “mail” as the local delivery agent, the Mlocal directive becomes:

```
Mlocal, P=/usr/sbin/amavis, F=lsDFMAw5:/|@qrn9, S=10/30, R=20/40,
      T=DNS/RFC822/X-Unix,
      A=amavis $f $u -- /bin/mail.local -d $u
```

The amavis script intervenes, handles the MIME, expands compressed attachments, runs them through the virus scanner, then returns them to the local mailer.

Finding and choosing a Unix-based virus scanner

Because most viruses affect Windows machines, the mainstream platform for virus scanning tools is, unsurprisingly, Windows. This leaves the Unix administrator with a slightly more difficult task of (a) finding a product that will run on his/her platform, (b) evaluating the quality of the product, and (c) getting knowledgeable support for the product.

To find a Unix-based virus-scanning product, consult the AMaViS web site. AMaViS is an active project and their documentation is kept fairly fresh. The list of supported

scanners in their documentation is a good place to start your search for a Unix-based scanner.

Evaluating Unix-based virus scanners is more difficult for the Unix administrator because the product reviews are almost exclusively for Windows-based products. There are two aspects to performance: 1) percentage of viruses caught, and 2) scanning speed. Several independent ratings organizations exist. Two recommendations are AV-Test.org and Virus Bulletin. Both organizations evaluate the effectiveness of the scanners (percentage of viruses caught). Although the tests are run on Windows machines, the best we can do is hope that similar algorithms are transferred to the Unix products. Regarding scanning speed, to my knowledge, there do not exist benchmark test results under Unix.

Finally, the administrator needs to find a good Unix implementation of the virus-scanning product. Most vendors showcase their Windows, Exchange, and Notes products, understandably, given the market share situation and vendor development expertise. Look for a vendor that appears to have Unix as a significant marketing target. Consider where their Unix products are featured in their Web sites. Ask the salesman a hard Unix implementation question and see if he can get a knowledgeable tech support person on the phone for pre-sales support.

References

AMaViS Team. "AMaViS - A Mail Virus Scanner." Aug. 13, 2001. http://www.amavis.org/ (Aug. 20, 2001).

Costales, Bryan. and Allman, Eric. Sendmail, Second Edition. Sebastopol. O'Reilly and Associates, Inc., 1997.

Daniels, PL. "SignatureDB – Digital Data Signature Database." http://www.pldaniels.com/signaturedb/index.html (Aug. 20, 2001).

Hardin, J. "Enhancing E-Mail Security With Procmail the E-mail Sanitizer" Aug. 12, 2001. http://www.impsec.org/email-tools/procmail-security.html (Aug. 20, 2001).

Kaspersky Lab. "Kaspersky Anti-Virus for Linux: User Guide."

Marx, A. "AV-Test.org - Tests of Anti-Virus Software: Current Tests." http://www.av-test.org/sites/tests.php3?lang=en (Aug. 20, 2001).

Sendmail Inc. "Filtering Mail with Sendmail." http://www.sendmail.com/partner/resources/development/milter_api/ (Aug. 20, 2001).

Trend Micro Inc. "Trend InterScan VirusWall 3.6: Administrator's Guide." Aug. 6, 2001. http://www.antivirus.com/download/documentation/internetgateway/files/isux36ag.pdf (Aug. 20, 2001).

Virus Bulletin. “Virus Bulletin 100% Awards.” http://www.virusbtn.com/100/ (Aug. 20, 2001).