Global Information Assurance Certification Paper

An Elementary Introduction to Sendmail

Jay Coleson
August 3, 2000

I. Introduction

It is known to some as "the buggiest daemon on Earth." [1] *Sendmail* is an agent by which email messages are transmitted and received over a network. Because it is most often included with Unix installations, it enjoys widespread acceptance as "the standard," but also undergoes widespread scrutiny. As with most applications typically bundled with a Unix kernel, sendmail is configured and operational "right out of the box." This can open many security vulnerabilities. But, as is also true about such Unix applications, a few simple modifications can greatly improve the security of the system. This paper will briefly touch on sendmail itself and some of its associated files, some of the security holes that have led to its infamous reputation, and finally, a few suggestions that will improve its security. Please note that the filenames and other details used in the following discussion are specific to Solaris 7. Other variants of Unix may well employ other conventions. The general principles presented, however, remain valid.

I. What is Sendmail?

A typical user tends to associate "email" with the user interface by which email messages are composed or read. The actual delivery and transport of email messages is much more complex than that. Tracing an email message through its creation, transportation, and delivery will impart a general understanding of the entire process.

To compose an email message, a *user agent* is employed. *User agents* have evolved to take the form of an email application with a *graphical user interface* (GUI), although a graphical interface is by no means necessary. The most recent versions of Solaris include agents with only a character interface such as `/usr/bin/mail` and `/usr/bin/mailx`, as well as some with GUIs such as `/usr/openwin/bin/mailtool` and `/usr/dt/bin/dtmail`. The writer's user agent passes the email message to a *message transfer agent* (MTA) to send the message over the network using the proper protocol.

> ***Sendmail*** is the most common MTA to be used with standard Unix systems. It uses the *Simple Mail Transfer Protocol* (SMTP) [2][3], communicating (by default) through the *well-known* TCP port number 25. [4]

In order to correctly route the email message through the network and to the proper recipient, the MTA must obtain the IP address corresponding to the email address listed in the message header. The MTA obtains such information from a *Domain Name Service* (DNS). [5][6] A DNS server keeps listings of domain names (which is the part of the email address to the right of the @) and the corresponding IP addresses. If these listings become corrupted, the email message can be routed incorrectly and can be used as part of an attack. (The interplay between an MTA and DNS is quite complex and can lead to many security vulnerabilities. [7][8] This paper will not delve into this interdependence, nor will it explore the various security issues involved.)

Upon arrival at the proper destination, the email message is passed by the reader's MTA to the reader's *delivery agent*. It is the job of the delivery agent to append the new email message to messages already stored. Solaris stores these email messages in the file `/var/mail/<userid>` and typically uses the delivery agent *mail.local*. The reader is then free to view the message using a user agent.

By default, sendmail is run as a *daemon* [1] process and is started during bootup. The `/etc/rc2.d/S<xx>sendmail` script ("`<xx>`" represents an integer) is called by the `/etc/rc2` boot script. It may also be started through a command line interface by runnning `/etc/init.d/sendmail` with the `start` option. In both cases, the start-up scripts invoke the sendmail binary, `/usr/lib/sendmail`, with the appropriate options, and reference the file `/etc/mail/sendmail.cf` for configuration information. Once up and running, the sendmail daemon simply responds to requests to send, or monitors to receive, SMTP traffic on TCP port 25.

II. Why should I be concerned about Sendmail?

Over the years, "the buggiest daemon on Earth" has earned its reputation, especially with regard to security issues. Many security holes in sendmail have been exploited as the bases of attacks since the Internet came to public prominence in the late 80's. Some of the security vulnerabilities amounted to nothing more than minor annoyances [9], some allowed unauthorized permissions [10][11], and still others could lead to full root access to the system. [12][13][14][15][16] The most famous of these is the *Internet Worm* attack [17] by Robert Morris, Jr. The internet worm attack exploited (among other things) a hole in the DEBUG mode of the sendmail code. Bugtraq [18] descibed the vulnerability by the following.

> Sendmail's debug mode allows the recipient of an email message to be a program that runs with the privileges of the user id which sendmail is running under. This user is normally root.
>
> This allows an attacker to set the recipient to as the shell and include shell command in the message body.

A more recent example is the so-called *MIME Conversion Buffer Overrun* attack. [19] By exploiting a bug in the default MIME conversion configuration of vulnerable sendmail versions and "sending a carefully crafted email message…, intruders may be able to force sendmail to execute arbitrary commands with root privileges." [20]

Sendmail also contained code bugs that could be exploited to render the system inoperable, i.e., produce a *denial of service* (DoS) attack. [21][22] One recent such attack exploited a particular *sleep()* call in the sendmail daemon code. By sending email messages timed to arrive at an interval less than that used in the *sleep()* call, the sendmail daemon would effectively stay "asleep" for the length of the attack There are even examples of *trojan horse* programs that use sendmail to allow the attacker to launch a shell on a remote host with root privileges. [23] . (See [24] for more exploits of all types.)

Granted, these examples are at least a few years old and represent holes most of which have long since been patched; however, many older versions of sendmail are still being used unpatched,

and are thus vulnerable to these and other exploits. Also, attacks against newer versions of sendmail will no doubt be found, given enough time and scrutiny.

III. Ok, so how do I secure Sendmail?

In general, the recommendations that apply to any Unix application apply to sendmail in particular. The first and foremost of these is "If you don't need it, don't run it!" Do not allow the sendmail daemon to come up and stay up if it is not necessary.

The most direct way to keep sendmail from coming up is to remove the sendmail binary, `/usr/lib/sendmail`. The sendmail start scripts could then also be removed. (In fact, the `/etc/rc2.d/S<xx>sendmail` script would have to be removed in order to keep the boot process from attempting to launch a non-existent process.)

If sendmail is needed, the daemon can still be prevented from coming up during the boot process by simply removing the `/etc/rc2.d/S<xx>sendmail` script or renaming it to something that does not begin with "`S`." (The `/etc/rc2` boot script searches for all files in the `/etc/rc2.d` directory that begin with "`S`" which are then interpreted as "startup" scripts. A filename in `/etc/rc2.d` that begins with "`K`" is likewise taken to be a "kill" script.) Use the `mv` command rather than a `cp-rm` combination to rename the script(s). These scripts are all hard-linked together. The `mv` command will maintain the links while a `cp-rm` combination will break them. If mail delivery via sendmail is necessary, it can be invoked directly by the user mail agent without the daemon remaining up. In order for this to occur, two modifications must be made. First, a *null-client* must be created in the `/etc/mail/sendmail.cf` configuration file. Second, sendmail must be run, at least, on a periodic basis to clear out the mail queue. This can easily be done from a *cron* job. (Consult [25] for a more complete description of this procedure.)

If it is determined that the sendmail daemon is to come up during the boot process and to remain up, e.g., the host is serving as a mail server, there are still several security precautions that can be taken.

1. Use local routers and firewalls to restrict access to TCP port 25 on the email host. This makes connecting to the sendmail daemon from an unauthorized host, at least, much more difficult. If access into the mail server is gained, sendmail can quite easily be used to launch attacks that further compromise the host (such as permitting root access) as well as to launch attacks against other hosts through email (such as *spamming*, *mail bombing*, DoS attacks, etc.)

2. "Lock down" all other processes running on this host. ("If you don't need it, don't run it!") If access is gained by *any* means, sendmail can quite easily be used in a malicious manner.

3. Run the most up-to-date version of sendmail possible with all relevant patches installed. The attacks against sendmail are extremely version-specific. The more recent versions have fewer holes and the application of all relevant patches insures that a minimum number of security holes exist.

4. Stay up-to-date with possible exploits by regularly visiting a website that is devoted to security issues, such as SecurityFocus (Bugtraq), PacketStorm, CIAC, CERT, etc., or by signing up for a mailing list that will send notification of possible exploits and/or needed fixes.

5. Change the SMTP login message (the sendmail *banner*) to omit version information. This is easily done by commenting out the default banner line and inserting another in the sendmail configuration file, `/etc/mail/sendmail.cf`. For Solaris 7, the banner message is stored in the variable `SmtpGreetingMessage`. There is no need to advertise the version of sendmail that is being run. Such advertisement makes a would-be attacker's job much easier.

6. Apply the proper ownership and permissions to the sendmail binary, the configuration file, the sendmail scripts, and other associated files and directories. Allow only what is absolutely necessary. This will help prevent a would-be attacker from altering the configuration to make access more easily gained. (See [26] and [27] for complete discussions of this issue with regards to recent versions of sendmail. For a good (but old – circa 1993) discussion of sendmail file permission issues, see [28].)

7. Consider using more secure replacements/additions for sendmail such as *SMAP* (for mail relays) and *smrsh*.

IV. Conclusion

Sendmail has earned its infamous reputation since the 1988 *Internet Worm* attack thrust it into the spotlight. The security holes found in sendmail since then have been numerous and dangerous; but, with increased scrutiny comes increased efforts to address these issues. In fact, the most recent versions exhibit very few known security holes. For the holes that remain and the holes that are yet to be found, the few simple suggestions presented above, suggestions that in reality apply to any standard Unix application, will maintain the sendmail service at a relatively secure level.

Specific References

[1] RaveN. *The Sendmal Tutorial*. Version 2.5. June 10, 2000.
URL: http://blacksun.box.sk/projects.php3 -> tutorials -> *The Sendmail Tutorial*

[2] Postal, Jonathan B. *Simple Mail Transfer Protocol* (RFC 821). Aug. 1982.
URL: http://www.ietf.org/rfc/rfc0821.txt

[3] Crocker, David H. *Standard for the Format of ARPA Internet Text Messages* (RFC 822). Aug. 13, 1982.
URL: http://www.ietf.org/rfc/rfc0822.txt

[4] IANA. *Port Numbers*. July 27, 2000
URL: http://www.isi.edu/in-notes/iana/assignments/port-numbers

[5] Mockapetris, P. *Domain Names – Concepts and Facilities* (RFC 1034). Nov. 1987.
URL: http://www.ietf.org/rfc/rfc1034.txt

[6] Mockapetris, P. *Domain Names – Implementation and Specification* (RFC 1035). Nov. 1987.
URL: http://www.ietf.org/rfc/rfc1035.txt

[7] CIAC. Alert G-14: *Domain Name Service Vulnerabilities*. Feb. 28, 1996.
URL: http://ciac.llnl.gov/ciac/bulletins/g-14.shtml

[8] Sun Microsystems. *sms.133.sendmail_bind_dns*. Packetstorm archives. March 8, 1996.
URL: http://packetstorm.securify.com/advisories/sms/sms.133.sendmail_bind_dns

[9] Smith, Simon. *sendmail 8.9.1-2.txt*. Packetstorm archives. Sept. 24, 1998.
URL: http://packetstorm.securify.com/new-exploits/sendmail.8.9.1-2.txt

[10] CERT. Advisory CA-96.25: *Sendmail Group Permissions Vulnerability*. Oct. 20, 1997.
URL: http://www.cert.org/advisories/CA-96.25.sendmail_groups.html

[11] CIAC. Alert H-11: *sendmail Group Permissions Vulnerability*. Dec. 10, 1996.
URL: http://ciac.llnl.gov/ciac/bulletins/h-11.shtml

[12] CERT. Advisory CA-96.24: *Sendmail Daemon Mode Vulnerability*. Sept. 24, 1997.
URL: http://www.cert.org/advisories/CA-96.24.sendmail.daemon/mode.html

[13] CERT. Advisory CA-96.20: *Sendmail Vulnerabilities*. Dec. 9, 1998.
URL: http://www.cert.org/advisories/CA-96.20.sendmail_vul.html

[14] CIAC. Alert F-28a: *Vulnerability in SunOS 4.1.* Sendmail (-oR option): Update*. Sept. 10, 1995.
URL: http://ciac.llnl.gov/ciac/bulletins/f-28a.shtml

[15] CIAC. Alert G-43a: *Vulnerabilities in Sendmail*. Dec. 5, 1997.
URL: http://ciac.llnl.gov/ciac/bulletins/g-43a.shtml

[16] CIAC. Alert H-07: *Sendmail SIGHUP-smtpd Vulnerability*. Nov. 22, 1996.
URL: http://ciac.llnl.gov/ciac/bulletins/h-07.shtml

[17] Page, Bob. *A Report on the Internet Worm*. Nov. 7, 1988.
URL: http://www.worm.net/page_worm.txt

[18] Bugtraq. *Berkeley Sendmail DEBUG Vulnerability* (bugtraq id 1). April 11, 2000
URL: http://www.securityfocus.com -> Vulnerabilities -> Database

[19] CIAC. Alert H-23: *Sendmail MIME Conversion Buffer Overrun Vulnerability*. Jan. 23, 1997.
URL: http://ciac.llnl.gov/ciac/bulletins/h-23.shtml

[20] CERT. Advisory CA-97.05: *MIME Conversion Buffer Overflow in Sendmail Versions 8.8.3 and 8.8.4*. Sept. 26, 1997. URL: http://www.cert.org/advisories/CA-97.05.sendmail.html

[21] Bugtraq. *Sendmail ETRN Denial of Service Vulnerability* (bugtraq id 904). April 11, 2000.
URL: http://www.securityfocus.com -> Vulnerabilities -> Database

[22] Zalewski, Michal. *sendmail-x.x.x-DoS.txt*. Packetstorm archives. Sept. 6, 1998.
URL: http://packetstorm.securify.com/new-exploits/sendmail-x.x.x-DoS.txt

[23] Naif. *sendmailcftrojan.tar.gz*. Packetstorm archives.
URL: http://packetstorm.securify.com/UNIX/penetration/rootkits/sendmailcftrojan.tar.gz

[24] Fadia, Ankit. *Various Sendmail Holes*. Version 1.0.
URL: http://blacksun.box.sk/projects.php3 -> tutorials -> *Various Sendmail Holes*

[25] Pomeranz, Hal. *Building Bastion Hosts with Solaris*. SANS: The Fifth Annual Conference on Securing UNIX and NT Systems. Oct. 3 – 10, 1999. URL: http://www.deer-run.com/solaris.html

[26] Sun Microsystems. *Sun/Solaris Vendor Specific Information*. URL: http://www.sendmail.org/sun-specific

[27] Sendmail.org. *Sendmail Frequently Asked Questions*. July 19, 2000 (Question 3.33 – May 24, 1999)
URL: http://www.sendmail.org/faq

[28] Costales, Bryan with Allman, Eric and Rickert, Neil. *Sendmail*. Sebastopol, CA: O'Reilly & Associates, Inc. 1993. Pp 235-240

General References

Gregory, Peter H. *SOLARIS Security*. Upper Saddle River, NJ: Printice Hall, 2000.

Stevens, W. Richard. *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, MA: Addison-Wesley, 1994.

Sunworld. *The Solaris Security FAQ* ("How do I secure sendmail?").
URL: http://www.sunworld.com/common/f_security-faq.html#Q2.24

Websites of interest

Computer Emergency Response Team (CERT) URL: http://www.cert.org

Computer Incident Advisory Capability (CIAC) group – part of the U.S. Dept. of Energy URL: http://ciac.llnl.gov

Harker sendmail reference page. URL: http://www.harker.com/sendmail/sendmail-ref.html

Internet Engineering Task Force (IETF) URL: http://www.ietf.org

PacketStorm URL: http://packetstorm.securify.com

SecurityFocus (home of Bugtraq) URL: http://www.securityfocus.com

Sendmail Consortium URL: http://www.sendmail.org

Last Updated: September 21st, 2017

Upcoming Training

SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS New York SEC401*	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague Summit & Training 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Mentor Session - SEC401	Minneapolis, MN	Oct 03, 2017 - Nov 14, 2017	Mentor
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
SANS Phoenix-Mesa 2017	Mesa, AZ	Oct 09, 2017 - Oct 14, 2017	Live Event
SANS Tysons Corner Fall 2017	McLean, VA	Oct 14, 2017 - Oct 21, 2017	Live Event
SANS Tokyo Autumn 2017	Tokyo, Japan	Oct 16, 2017 - Oct 28, 2017	Live Event
CCB Private SEC401 Oct 17	Brussels, Belgium	Oct 16, 2017 - Oct 21, 2017	
SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201710,	Oct 23, 2017 - Nov 29, 2017	vLive
Community SANS Omaha SEC401	Omaha, NE	Oct 23, 2017 - Oct 28, 2017	Community SANS
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
San Diego Fall 2017 - SEC401: Security Essentials Bootcamp Style	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS Gulf Region 2017	Dubai, United Arab Emirates	Nov 04, 2017 - Nov 16, 2017	Live Event
SANS Miami 2017	Miami, FL	Nov 06, 2017 - Nov 11, 2017	Live Event
Community SANS Vancouver SEC401*	Vancouver, BC	Nov 06, 2017 - Nov 11, 2017	Community SANS
Community SANS Colorado Springs SEC401**	Colorado Springs, CO	Nov 06, 2017 - Nov 11, 2017	Community SANS
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
SANS Sydney 2017	Sydney, Australia	Nov 13, 2017 - Nov 25, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Nov 27, 2017 - Dec 02, 2017	Community SANS
SANS London November 2017	London, United Kingdom	Nov 27, 2017 - Dec 02, 2017	Live Event
SANS San Francisco Winter 2017	San Francisco, CA	Nov 27, 2017 - Dec 02, 2017	Live Event
Community SANS St. Louis SEC401	St Louis, MO	Nov 27, 2017 - Dec 02, 2017	Community SANS
SANS Khobar 2017	Khobar, Saudi Arabia	Dec 02, 2017 - Dec 07, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Dec 04, 2017 - Dec 09, 2017	Community SANS
SANS Munich December 2017	Munich, Germany	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Austin Winter 2017	Austin, TX	Dec 04, 2017 - Dec 09, 2017	Live Event