

[UNIX] GNU Mailutils Multiple Vulnerabilities (Buffer Overflows, Format String, DoS)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-05/0141.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 05/26/05

To: list@securiteam.com

Date: 26 May 2005 18:04:26 +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>

GNU Mailutils Multiple Vulnerabilities (Buffer Overflows, Format String, DoS)

SUMMARY

"GNU <<http://www.gnu.org/software/mailutils/>> Mailutils is a collection of mail-related utilities. At the core of Mailutils is libmailbox, a library which provides access to various forms of mailbox files (including remote mailboxes via popular protocols). It also provides support for parsing of RFC-822 style messages and MIME support."

Buffer overflow and a format string vulnerabilities allow attackers to execute arbitrary code from remote using the GNU Mailutils programs. A denial of service vulnerability allows attackers to stop the server running GNU Mailutils from responding.

DETAILS

Vulnerable Systems:

- * GNU Mailutils version 0.6
- * GNU Mailutils version 0.5

Immune Systems:

Securiteam: [UNIX] GNU Mailutils Multiple Vulnerabilities (Buffer Overflows, Format String, DoS)

* GNU Mailutils version 0.6.90

Format String:

Remote exploitation of a format string vulnerability in the imap4d server allow an unauthenticated attacker to execute arbitrary code.

The imap4d server allows remote users to retrieve their email via the Internet Message Access Protocol, Version 4rev1 as specified in RFC3501. This is a client/server protocol supported by a large number of email clients on multiple platforms.

The vulnerability specifically exists in the handling of the command tag supplied by the remote user. Each client command sent to the server is prefixed with an identifier which is typically a short alphanumeric string such as "A0001". A different tag is generated by the client for each command. When the server has completed the task, with either success or failure, the server will send a reply with the same tag.

Code Snips:

```
asprintf (&tempbuf, "%s %s%s %s\r\n", command->tag, sc2string (rc),
         command->name, format);
va_start (ap, format);
vasprintf (&buf, tempbuf, ap);
```

The asprintf() command allocates a new string, created by joining the values of the tag supplied by the remote user, the text version of the result code, the name of the command being executed, and the original format string supplied to this function.

The effect of this line is to generate a new format string string which is used to generate the output. As there is no check for format specifiers in the user supplied input, a remotely exploitable condition occurs.

Successful exploitation allows remote unauthenticated attackers to execute arbitrary commands on an affected system as the 'daemon' user.

Sending the following command to an affected server will cause the current connection to die when the fork()ed instance of the server crashes:

```
%n%n%n%n%n die.
```

The '%n' format specifier writes the number of characters in the output string generated so far to the memory address pointed at by the current argument. In this case, the process attempts to write the value 0 to the next 5 memory locations in the argument list. As some of these arguments are not valid pointers, the server dies attempting to write to an invalid memory location. This will not cause a denial of service, as a new instance of the server is spawned for each accepted connection.

Information about the values on the stack below the current position can be gained by sending a string similar to:

```
%p-%p-%p-%p-%p-%p-%p-%p-%p-%p info
```

Using these format specifiers it is possible to construct a sequence of commands which will cause arbitrary values to be written to arbitrary locations, allowing the execution of arbitrary code.

FETCH Command Resource Consumption DoS:

Remote exploitation of an input validation in the FETCH command of the imap4d server allow an authenticated remote attacker to perform a denial of service against an affected system.

The vulnerability specifically exists in the handling of the sequence range argument to the FETCH command. A sequence range such as 1:4294967294 will cause the the spawned instance of the server to enter what is effectively an infinite loop, allocating memory on each cycle.

Successful exploitation of the vulnerability would allow a remote authenticated user to cause the system hosting the imap4d to become unresponsive due to lack of memory resources. On some operating systems, such as Linux, processes will be killed by the kernel out of memory manager.

imap4d fetch_io Heap overflow:

Remote exploitation of an integer overflow in the fetch_io function of the imap4d server allow an authenticated remote attacker to execute arbitrary code.

The vulnerability specifically exists in the handling of partial message requests. By supplying a value for the 'END' parameter equal to 2 less than the largest value an integer on the affected system can hold, it is possible to cause the server to allocate a much smaller chunk of memory. An overflow can occur when this memory chunk is referenced.

Successful exploitation of this vulnerability could allow a remote authenticated user to execute arbitrary commands in the context of the 'daemon' user. As the service is forked from a parent process, it would be possible for a remote attacker to attempt to exploit this vulnerability multiple times, although each failed attempt which caused a crash would be logged.

header_get_field_name() Buffer Overflow:

Exploitation of a buffer overflow vulnerability in the mail binary of the GNU Projects Mailutils package may allow a remote attacker to execute commands with the privileges of the targeted user.

Due to a coding error in the library function header_get_field_name() in mailbox/header.c, a buffer overflow condition exists. The code below checks if the value of 'len' is greater than the value of 'buflen', but always sets the value of 'len' to be the same, effectively performing no instruction.

Securiteam: [UNIX] GNU Mailutils Multiple Vulnerabilities (Buffer Overflows, Format String, DoS)

```
len = (len > buflen) ? len : len;
```

This code will set 'len' to the value of 'buflen' if 'len' is greater than 'buflen'. This typo allows the buffer overflow to occur. The code should be:

```
len = (len > buflen) ? buflen : len;
```

Successful exploitation of the vulnerability would allow an email sent by a remote user to cause a buffer overflow, allowing execution of arbitrary commands in the context of the targeted user. Access to a user account may allow further escalation of privileges via local attacks.

Workaround:

When possible, run client software as a regular user with limited access to system resources. This may limit the immediate consequences of client-side vulnerabilities.

CVE Information:

<<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-1520>>

CAN-2005-1520

<<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-1521>>

CAN-2005-1521

<<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-1522>>

CAN-2005-1522

<<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-1523>>

CAN-2005-1523

Disclosure Timeline:

05/12/2005 – Initial vendor notification

05/12/2005 – Initial vendor response

05/25/2005 – Public disclosure

ADDITIONAL INFORMATION

The information has been provided by

<<mailto:idlabs-advisories@idefense.com>> idlabs.

The original article can be found at:

<<http://www.idefense.com/application/poi/display?id=246&type=vulnerabilities>>

<http://www.idefense.com/application/poi/display?id=246&type=vulnerabilities>,

<<http://www.idefense.com/application/poi/display?id=247&type=vulnerabilities>>

<http://www.idefense.com/application/poi/display?id=247&type=vulnerabilities>,

<<http://www.idefense.com/application/poi/display?id=248&type=vulnerabilities>>

<http://www.idefense.com/application/poi/display?id=248&type=vulnerabilities>,

<<http://www.idefense.com/application/poi/display?id=249&type=vulnerabilities>>

<http://www.idefense.com/application/poi/display?id=249&type=vulnerabilities>

=====

Securiteam: [UNIX] GNU Mailutils Multiple Vulnerabilities (Buffer Overflows, Format String, DoS)

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.