

# [UNIX] Asterisk Multiple RTP Buffer Overflows

---

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2008-03/msg00069.html>

---

- *From:* SecuriTeam <[support@xxxxxxxxxxxxxx](mailto:support@xxxxxxxxxxxxxx)>
  - *Date:* 19 Mar 2008 09:09:58 +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>

-----

Asterisk Multiple RTP Buffer Overflows

---

## SUMMARY

<<http://www.asterisk.org/>> Asterisk is "an open source telephony engine and toolkit. Asterisk implements the Session Initiation Protocol (SIP)". The Mu Security Research team has found two security issues in the SDP parser in Asterisk 1.4.18. One is an invalid write to an attacker-controllable, almost arbitrary memory location and the other is a stack buffer overflow with limited attacker-controllable values.

## DETAILS

Vulnerable Systems:

\* Asterisk version 1.4.18.0 and prior

Immune Systems:

\* Asterisk version 1.4.18.1

1) Sending an invalid RTP payload type number in the SDP payload of an INVITE message can cause a write to an invalid memory location. An attacker would have some control over the memory location.

The invalid memory write is in `ast_rtp_unset_m_type()` (main/rtp.c, line

## [UNIX] Asterisk Multiple RTP Buffer Overflows

1655) called by process\_line() (channels/chan\_sip.c, line 5275).  
ast\_rtp\_unset\_mt\_type() does not validate pt, while it is validated in  
ast\_rtp\_set\_mt\_type() (line 1642). The attacker controls pt and could  
write a 0 to a wide range of memory locations.

Example invalid SDP payload (invalid RTP payload type is 780903144):

```
v=0
o=- 817933771 817933775 IN IP4 10.10.1.101
s=session-name
c=IN IP4 10.10.1.101
t=0 0
m=audio 5000 RTP/AVP 0
a=rtpmap:780903144 PCMU/8000
a=rtpmap:4 G723/8000/1
a=rtpmap:97 telephone-event/8000
```

2) Sending more than 32 RTP payload type number attributes in the SDP payload of a SIP INVITE will overflow a buffer on the stack. An attacker would have some control over the values written.

In process\_sdp() (channels/chan\_sip.c, line 4980), rtpmap codecs are stored in found\_rtpmap\_codecs, an array of 32 ints. The number of codecs in the map is stored in last\_rtpmap\_codec. Codecs are appended to the array without checking the size of the array (line 5258). Up to 64 (SIP\_MAX\_LINES). An attacker would have some control over the values written – the codec must be between 0 and 256 (MAX\_RTP\_PT).

Example SDP payload:

```
v=0
o=- 817933771 817933775 IN IP4 10.10.1.101
s=session-name
c=IN IP4 10.10.1.101
t=0 0
m=audio 5000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
[... repeat this line ...]
a=rtpmap:4 G723/8000/1
a=rtpmap:97 telephone-event/8000
```

Vendor Response / Solution:

Fixed in Asterisk 1.4.18.1 and other branches. Available from  
<http://www.asterisk.org>

History:

March 11, 2008 – First contact with vendor  
March 18, 2008 – Vendor releases fix and advisory

CVE Information:

<<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1289>>  
CVE-2008-1289

[UNIX] Asterisk Multiple RTP Buffer Overflows

ADDITIONAL INFORMATION

The information has been provided by Mu Security.

The original article can be found at:

<<http://labs.musecurity.com/advisories/MU-200803-01.txt>>

<http://labs.musecurity.com/advisories/MU-200803-01.txt>

=====

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@xxxxxxxxxxxxxxxx

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxxxx

=====

=====

**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.