Bug 22542 (CVE-2022-23219) - buffer overflow in sunrpc clnt_create (CVE-2022-23219)

Status:	RESOLVED FIXED	**Reported:**	2017-12-03 22:27 UTC by Martin Sebor
Alias:	CVE-2022-23219	**Modified:**	2022-01-17 13:07 UTC (History)
		CC List:	7 users (show)
Product:	glibc		
Component:	network (show other bugs)	**See Also:**	~~CVE-2022-23218~~
Version:	2.24	**Host:**	
		Target:	
Importance:	P2 normal	**Build:**	
Target Milestone:	2.35	**Last reconfirmed:**	2018-02-06 00:00:00
Assignee:	Florian Weimer		
		Flags:	fweimer: security+
URL:			
Keywords:			
Depends on:			
Blocks:			

Attachments

Add an attachment (proposed patch, testcase, etc.)

Note

You need to log in before you can comment on or make changes to this bug.

Martin Sebor 2017-12-03 22:27:53 UTC **Description**

The clnt_create() function calls strcpy() to copy the string pointed to by the hostname argument to the sun_addr member array of a struct sockaddr_un object allocated on the stack. When the string is longer than fits in the array the function corrupts the calling process' stack due to the buffer overflow.

I noticed this while developing the patch suggested here: https://sourceware.org/ml/libc-alpha/2017-11/msg00932.html.

```
$ cat d.c && gcc -Wall d.c && valgrind ./a.out

#include <errno.h>
#include <rpc/clnt.h>
#include <sys/socket.h>
#include <sys/un.h>
#include <string.h>

int main ()
{
  char name [sizeof ((struct sockaddr_un*)0)->sun_path * 2];
  memset (name, 'x', sizeof name - 1);
  name [sizeof name - 1] = '\0';

  CLIENT *clnt = clnt_create (name, 0, 0, "unix");
```

```
  if (clnt)
    clnt_destroy (clnt);
}
==18499== Memcheck, a memory error detector
==18499== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et al.
==18499== Using Valgrind-3.12.0 and LibVEX; rerun with -h for copyright info
==18499== Command: ./a.out
==18499==
==18499== Source and destination overlap in strcpy(0xffefffcc2, 0xffefffd70)
==18499==    at 0x4C30E06: __GI_strcpy (vg_replace_strmem.c:507)
==18499==    by 0x4F6FE30: clnt_create (in /usr/lib64/libc-2.24.so)
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878780077: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==
==18499== Jump to the invalid address stated on the next line
==18499==    at 0x7878787878787878: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878780077: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==  Address 0x7878787878787878 is not stack'd, malloc'd or (recently) free'd
==18499==
==18499==
==18499== Process terminating with default action of signal 11 (SIGSEGV)
==18499==  Bad permissions for mapped region at address 0x7878787878787878
==18499==    at 0x7878787878787878: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878780077: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==    by 0x7878787878787877: ???
==18499==
==18499== HEAP SUMMARY:
==18499==     in use at exit: 0 bytes in 0 blocks
==18499==   total heap usage: 2 allocs, 2 frees, 272 bytes allocated
==18499==
==18499== All heap blocks were freed -- no leaks are possible
==18499==
==18499== For counts of detected and suppressed errors, rerun with: -v
==18499== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
Segmentation fault (core dumped)
```

Martin Sebor 2017-12-03 23:10:24 UTC [Comment 1]

Patch posted for review:
https://sourceware.org/ml/libc-alpha/2017-12/msg00058.html

Florian Weimer 2022-01-17 13:06:49 UTC **Comment 2**

Fixed for glibc 2.35 via:

```
commit 226b46770c82899b555986583294b049c6ec9b40
Author: Florian Weimer <fweimer@redhat.com>
Date:   Mon Jan 17 10:21:34 2022 +0100

    CVE-2022-23219: Buffer overflow in sunrpc clnt_create for "unix" (~~bug 22542~~)

    Processing an overlong pathname in the sunrpc clnt_create function
    results in a stack-based buffer overflow.

    Reviewed-by: Siddhesh Poyarekar <siddhesh@sourceware.org>
```

Florian Weimer 2022-01-17 13:07:03 UTC **Comment 3**

.