New issue

Jump to bottom

stack_based buffer #219

Open x00x00x00x00 opened this issue on May 27, 2021 · 0 comments

x00x00x00x00 commented on May 27, 2021 • edited

Hi Team,

Stack based buffer overflow is observed in read_write.c and gattlib.c while fuzzing **GATTLIB (releases - master, v0.3-rc1, dev, various-fixes)** using **CLANG** with **AFL FUZZER**

Vulnerable code from read_write.c -

```
connection = gattlib_connect(NULL, argv[1], GATTLIB_CONNECTION_OPTIONS_LEGACY_DEFAULT);
if (connection == NULL) {
        fprintf(stderr, "Fail to connect to the bluetooth device.\n");
        return 1;
}
```

Vulnerable code from gattlib.c -

```
get_device_path_from_mac(adapter_name, dst, object_path, sizeof(object_path));

// Transform string from 'DA:94:40:95:E0:87' to 'dev_DA_94_40_95_E0_87'
strncpy(device_address_str, mac_address, sizeof(device_address_str));
for (int i = 0; i < strlen(device_address_str); i++) {
if (device_address_str[i] == ':') {
device_address_str[i] = '_';
}
}
```

Steps to Reproduce -

cd
mkdir build && cd build

cmake .. -DCMAKE_CXX_FLAGS="-fsanitize=address -fsanitize=leak -g" -DCMAKE_C_FLAGS="-fsanitize=address -fsanitize=leak -g"

make

./examples/read_write/read_write 11111111111111111111111111111111 read 00002a29-0000-1000-8000-00805f9b34fb

ASAN output -

```
=================================================================
==72493==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7ffc83cd4d95 at pc 0x00000042efb9 bp 0x7ffc83cd4d50 sp 0x7ffc83cd4510
READ of size 22 at 0x7ffc83cd4d95 thread T0
-0 0x42efb8 in strlen (/home/zero/newfuz/gattlib/build/examples/read_write/read_write+0x42efb8)
-1 0x7fb3c462c532 in get_device_path_from_mac /home/zero/newfuz/gattlib/dbus/gattlib.c:114:22
-2 0x7fb3c462c532 in gattlib_connect /home/zero/newfuz/gattlib/dbus/gattlib.c:150:2
-3 0x4c471b in main /home/zero/newfuz/gattlib/examples/read_write/read_write.c:71:15
-4 0x7fb3c394e0b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/csu/../csu/libc-start.c:308:16
-5 0x41c38d in _start (/home/zero/newfuz/gattlib/build/examples/read_write/read_write+0x41c38d)

Address 0x7ffc83cd4d95 is located in stack of thread T0 at offset 53 in frame
-0 0x7fb3c462c2bf in gattlib_connect /home/zero/newfuz/gattlib/dbus/gattlib.c:136
```

This frame has 3 object(s):
[32, 53) 'device_address_str.i' (line 103) <== Memory access at offset 53 overflows this variable
[96, 104) 'error' (line 140)
[128, 228) 'object_path' (line 141)
HINT: this may be a false positive if your program uses some custom stack unwind mechanism, swapcontext or vfork
(longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow (/home/zero/newfuz/gattlib/build/examples/read_write/read_write+0x42efb8) in strlen
Shadow bytes around the buggy address:
0x100010792960: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x100010792970: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x100010792980: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x100010792990: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000107929a0: 00 00 00 00 00 00 00 00 00 00 00 00 f1 f1 f1 f1
=>0x1000107929b0: 00 00[05]f2 f2 f2 f2 f2 00 f2 f2 f2 00 00 00 00
0x1000107929c0: 00 00 00 00 00 00 00 00 04 f3 f3 f3 f3 f3 f3 f3
0x1000107929d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000107929e0: 00 00 00 00 00 00 00 00 00 00 00 00 f1 f1 f1 f1
0x1000107929f0: 00 f2 f2 f2 f8 f2 f2 f2 f8 f8 f8 f8 f2 f2 f2
0x100010792a00: f2 f2 f8 f8 f8 f8 f3 f3 f3 f3 00 00 00 00
Shadow byte legend (one shadow byte represents 8 application bytes):
Addressable: 00
Partially addressable: 01 02 03 04 05 06 07
Heap left redzone: fa
Freed heap region: fd
Stack left redzone: f1
Stack mid redzone: f2
Stack right redzone: f3
Stack after return: f5
Stack use after scope: f8
Global redzone: f9
Global init order: f6
Poisoned by user: f7
Container overflow: fc
Array cookie: ac
Intra object redzone: bb
ASan internal: fe
Left alloca redzone: ca
Right alloca redzone: cb
Shadow gap: cc
==72493==ABORTING

Request team to implement proper patch and validate

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant