Heap out of bound read in Gawk

Description

This vulnerability is of type Heap out of bound read. To be detailed, the array "the_args" takes an unsafe index "val", while it does not validate the index to ensure the index refers to a valid position in the array, such as exceedingly large or negative.The bug exists in latest stable release (gawk-5.1.1) and latest master branch (12f0f4f27872cd4df6c63977fd0c6ff63d29e424, updated in July 29, 2022). Specifically, the vulnerable code (located at builtin.c) and the bug's basic explanation are highlighted as follows:

```
// builtin.c
    } else if (! do_traditional && isdigit((unsigned char) *s1)) {
        int val = 0;

        for (; n0 > 0 && *s1 && isdigit((unsigned char) *s1); s1++, n0--) {
            val *= 10;
            val += *s1 - '0';
        }
        if (*s1 != '$') {
            msg(_("fatal: no `$' supplied for positional field width or
precision"));
            goto out;
        } else {
            s1++;
            n0--;
        }
        if (val >= num_args) {
            toofew = true;
            break;
        }
    // line 970
    // under this poc, the untrusted index is used to access the array.
        arg = the_args[val];
    } else {
```

Proof of Concept

Build the gawk (5.1.1 or latest commit 12f0f4f27872cd4df6c63977fd0c6ff63d29e424) and run it using the input POC.

```
# build the gawk with address sanitizer
export CFLAGS=" -fsanitize=address "; export CXXFLAGS=" -fsanitize=address ";
export LDFLAGS=" -fsanitize=address ";
CFGARG=" --enable-shared=no "; configure

AFL_USE_ASAN=1 LD=afl-gcc--disable-shared make

# the INPUT_FILE is not decesive, which can be any valid input.
./gawk -f POC_FILE INPUT_FILE
```

The stack dump is:

```
=================================================================
==15688==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x604000000088
at pc 0x5555556f2aae bp 0x7fffffffa8d0 sp 0x7fffffffa8c8
READ of size 8 at 0x604000000088 thread T0
    #0 0x5555556f2aad  (/src/Source/gawk-5.1.1/gawk+0x19eaad) (BuildId:
cff2b462aff47cf214070164e24ef91fbd66c30f)
    #1 0x555555702506  (/src/Source/gawk-5.1.1/gawk+0x1ae506) (BuildId:
cff2b462aff47cf214070164e24ef91fbd66c30f)
    #2 0x55555570327b  (/src/Source/gawk-5.1.1/gawk+0x1af27b) (BuildId:
cff2b462aff47cf214070164e24ef91fbd66c30f)
    #3 0x555555791b77  (/src/Source/gawk-5.1.1/gawk+0x23db77) (BuildId:
cff2b462aff47cf214070164e24ef91fbd66c30f)
    #4 0x55555580f443  (/src/Source/gawk-5.1.1/gawk+0x2bb443) (BuildId:
cff2b462aff47cf214070164e24ef91fbd66c30f)
    #5 0x7ffff797ed8f  (/lib/x86_64-linux-gnu/libc.so.6+0x29d8f) (BuildId:
69389d485a9793dbe873f0ea2c93e02efaa9aa3d)
    #6 0x7ffff797ee3f  (/lib/x86_64-linux-gnu/libc.so.6+0x29e3f) (BuildId:
69389d485a9793dbe873f0ea2c93e02efaa9aa3d)
    #7 0x5555555cdde4  (/src/Source/gawk-5.1.1/gawk+0x79de4) (BuildId:
cff2b462aff47cf214070164e24ef91fbd66c30f)

0x604000000088 is located 8 bytes to the left of 40-byte region
[0x604000000090,0x6040000000b8)
allocated by thread T0 here:
    #0 0x555555650c2e  (/src/Source/gawk-5.1.1/gawk+0xfcc2e) (BuildId:
cff2b462aff47cf214070164e24ef91fbd66c30f)
    #1 0x5555556cd7e2  (/src/Source/gawk-5.1.1/gawk+0x1797e2) (BuildId:
cff2b462aff47cf214070164e24ef91fbd66c30f)
    #2 0x5555556dcb72  (/src/Source/gawk-5.1.1/gawk+0x188b72) (BuildId:
cff2b462aff47cf214070164e24ef91fbd66c30f)
    #3 0x55555580eef4  (/src/Source/gawk-5.1.1/gawk+0x2baef4) (BuildId:
cff2b462aff47cf214070164e24ef91fbd66c30f)
    #4 0x7ffff797ed8f  (/lib/x86_64-linux-gnu/libc.so.6+0x29d8f) (BuildId:
69389d485a9793dbe873f0ea2c93e02efaa9aa3d)

SUMMARY: AddressSanitizer: heap-buffer-overflow (/src/Source/gawk-
5.1.1/gawk+0x19eaad) (BuildId: cff2b462aff47cf214070164e24ef91fbd66c30f)
Shadow bytes around the buggy address:
  0x0c087fff7fc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c087fff7fd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c087fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c087fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c087fff8000: fa fa 00 00 00 00 00 fa fa fa 00 00 00 00 06 fa
=>0x0c087fff8010: fa[fa]00 00 00 00 00 fa fa fa fd fd fd fd fd fa
  0x0c087fff8020: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c087fff8030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c087fff8040: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c087fff8050: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c087fff8060: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
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
==15688==ABORTING

#0  0x00005555556f2ab5 in format_tree (
    fmt_string=0x611000000900 '4' <repeats 28 times>,
"$3Urint\177/\033/8Y))9=$uir!nexu^%3*0$ttt*F~$9=$uiM\333RSH\003r64\377\063*\033/$
{Qr\377\377\377\177/\033/8Y))*,rfor!nexu^%3*", '4' <repeats 19 times>,
"$3Urint\177/\033/8Y))9=$uir!nexu^%3*0$ttt*F~$9=$uiM\333RSH\003r64b3*\033/$3in$99
"..., n0=108, the_args=0x6030000003d0, num_args=1) at builtin.c:970
#1  0x0000555555702507 in printf_common (nargs=1) at builtin.c:1682
#2  0x000055555570327c in do_printf (nargs=1, redirtype=0)
    at builtin.c:1758
#3  0x0000555555791b78 in r_interpret (code=0x622000001908)
    at ./interpret.h:1076
#4  0x000055555580f444 in main (argc=4, argv=0x7fffffffe3d8)
    at main.c:526
```

Impact

The bug triggered by the heap out of bound read would cause the crash of the software and might be used by attackers to steal sensitive information in memory, such as secret addresses. Besides, it is possible for the attacker to exploit other weak points to denial of service or launch remote code execution.

Reference

POC FILE