index : coreutils.git

GNU coreutils

master switch

summary refs log tree commit diff log msg search

author	Pádraig Brady <P@draigBrady.com>	2025-05-20 16:03:44 +0100
committer	Pádraig Brady <P@draigBrady.com>	2025-05-20 19:36:00 +0100
commit	8c9602e3a145e9596dc1a63c6ed67865814b6633 (patch)	
tree	6a9cf76e9963f848b3f0854ebedd504c47a749e5	
parent	44695761546d8f8b3dadd6d4e56a47b8ae69acd7 (diff)	

diff options

context: 3
space: include
mode: unified

sort: fix buffer under-read (CWE-127)

```
* src/sort.c (begfield): Check pointer adjustment
to avoid Out-of-range pointer offset (CWE-823).
(limfield): Likewise.
* tests/sort/sort-field-limit.sh: Add a new test,
which triggers with ASAN or Valgrind.
* tests/local.mk: Reference the new test.
* NEWS: Mention bug fix introduced in v7.2 (2009).
Fixes https://bugs.gnu.org/78507
```

Diffstat

-rw-r--r--	NEWS	5
-rw-r--r--	src/sort.c	12
-rw-r--r--	tests/local.mk	1
-rwxr-xr-x	tests/sort/sort-field-limit.sh	35

4 files changed, 51 insertions, 2 deletions

```
diff --git a/NEWS b/NEWS
index 6ff403206..923aa72f8 100644
--- a/NEWS
+++ b/NEWS
@@ -8,6 +8,11 @@ GNU coreutils NEWS                                    -*- outline -*-
   copying to non-NFS files from NFSv4 files with trivial ACLs.
   [bug introduced in coreutils-9.6]

+  sort with key character offsets of SIZE_MAX, could induce
+  a read of 1 byte before an allocated heap buffer. For example:
+  'sort +0.18446744073709551615R input' on 64 bit systems.
+  [bug introduced in coreutils-7.2]
+

  * Noteworthy changes in release 9.7 (2025-04-09) [stable]


diff --git a/src/sort.c b/src/sort.c
index b10183b6f..7af1a2512 100644
--- a/src/sort.c
+++ b/src/sort.c
@@ -1644,7 +1644,11 @@ begfield (struct line const *line, struct keyfield const *key)
       ++ptr;

   /* Advance PTR by SCHAR (if possible), but no further than LIM.  */
```

```
-  ptr = MIN (lim, ptr + schar);
+  size_t remaining_bytes = lim - ptr;
+  if (schar < remaining_bytes)
+    ptr += schar;
+  else
+    ptr = lim;

   return ptr;
 }
@@ -1746,7 +1750,11 @@ limfield (struct line const *line, struct keyfield const *key)
           ++ptr;

       /* Advance PTR by ECHAR (if possible), but no further than LIM.  */
-      ptr = MIN (lim, ptr + echar);
+      size_t remaining_bytes = lim - ptr;
+      if (echar < remaining_bytes)
+        ptr += echar;
+      else
+        ptr = lim;
     }

   return ptr;

diff --git a/tests/local.mk b/tests/local.mk
index 4da6756ac..642d225fa 100644
--- a/tests/local.mk
+++ b/tests/local.mk
@@ -388,6 +388,7 @@ all_tests =                                          \
   tests/sort/sort-debug-keys.sh                      \
   tests/sort/sort-debug-warn.sh                      \
   tests/sort/sort-discrim.sh                         \
+  tests/sort/sort-field-limit.sh                     \
   tests/sort/sort-files0-from.pl                     \
   tests/sort/sort-float.sh                           \
   tests/sort/sort-h-thousands-sep.sh                 \

diff --git a/tests/sort/sort-field-limit.sh b/tests/sort/sort-field-limit.sh
new file mode 100755
index 000000000..52d8e1d17
--- /dev/null
+++ b/tests/sort/sort-field-limit.sh
@@ -0,0 +1,35 @@
+#!/bin/sh
+# From 7.2-9.7, this would trigger an out of bounds mem read
+
+# Copyright (C) 2025 Free Software Foundation, Inc.
+
+# This program is free software: you can redistribute it and/or modify
+# it under the terms of the GNU General Public License as published by
+# the Free Software Foundation, either version 3 of the License, or
+# (at your option) any later version.
+
+# This program is distributed in the hope that it will be useful,
+# but WITHOUT ANY WARRANTY; without even the implied warranty of
+# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
+# GNU General Public License for more details.
+
+# You should have received a copy of the GNU General Public License
+# along with this program.  If not, see <https://www.gnu.org/licenses/>.
+
+. "${srcdir=.}/tests/init.sh"; path_prepend_ ./src
+print_ver_ sort
+getlimits_
+
+# This issue triggers with valgrind or ASAN
```

```
+valgrind --error-exitcode=1 sort --version 2>/dev/null &&
+  VALGRIND='valgrind --error-exitcode=1'
+
+{ printf '%s\n' aa bb; } > in || framework_failure_
+
+_POSIX2_VERSION=200809 $VALGRIND sort +0.${SIZE_MAX}R in > out || fail=1
+compare in out || fail=1
+
+_POSIX2_VERSION=200809 $VALGRIND sort +1 -1.${SIZE_MAX}R in > out || fail=1
+compare in out || fail=1
+
+Exit $fail
```