



author Vasiliy Kovalev <kovalev@altlinux.org> 2024-10-19 22:13:03 +0300
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2025-04-25 10:47:52 +0200
commit [84e8719c087e68c967975b78e67be54f697c957f](#) (patch)
tree [3774edbd1c30df3935d0f85e8f1260abedb3a61d](#)
parent [81326828560092be2a1858dd8088d1fdb002b618](#) (diff)
download [linux-84e8719c087e68c967975b78e67be54f697c957f.tar.gz](#)

diff options

context:

3 ▼

space:

include ▼

mode:

unified ▼

hfs/hfsplus: fix slab-out-of-bounds in hfs_bnode_read_key

commit [bb5e07cb927724e0b47be371fa081141cfb14414](#) upstream.

Syzbot reported an issue in hfs subsystem:

BUG: KASAN: slab-out-of-bounds in memcpy_from_page include/linux/highmem.h:423 [inline]
BUG: KASAN: slab-out-of-bounds in hfs_bnode_read fs/hfs/bnode.c:35 [inline]
BUG: KASAN: slab-out-of-bounds in hfs_bnode_read_key+0x314/0x450 fs/hfs/bnode.c:70
Write of size 94 at addr ffff8880123cd100 by task syz-executor237/5102

Call Trace:

<TASK>
__dump_stack lib/dump_stack.c:94 [inline]
dump_stack_lvl+0x241/0x360 lib/dump_stack.c:120
print_address_description mm/kasan/report.c:377 [inline]
print_report+0x169/0x550 mm/kasan/report.c:488
kasan_report+0x143/0x180 mm/kasan/report.c:601
kasan_check_range+0x282/0x290 mm/kasan/generic.c:189
__asan_memcpy+0x40/0x70 mm/kasan/shadow.c:106
memcpy_from_page include/linux/highmem.h:423 [inline]
hfs_bnode_read fs/hfs/bnode.c:35 [inline]
hfs_bnode_read_key+0x314/0x450 fs/hfs/bnode.c:70
hfs_brec_insert+0x7f3/0xbd0 fs/hfs/brec.c:159
hfs_cat_create+0x41d/0xa50 fs/hfs/catalog.c:118
hfs_mkdir+0x6c/0xe0 fs/hfs/dir.c:232
vfs_mkdir+0x2f9/0x4f0 fs/namei.c:4257
do_mkdirat+0x264/0x3a0 fs/namei.c:4280
__do_sys_mkdir fs/namei.c:4300 [inline]
__se_sys_mkdir fs/namei.c:4298 [inline]
__x64_sys_mkdir+0x6c/0x80 fs/namei.c:4298
do_syscall_x64 arch/x86/entry/common.c:52 [inline]
do_syscall_64+0xf3/0x230 arch/x86/entry/common.c:83
entry_SYSCALL_64_after_hwframe+0x77/0x7f
RIP: 0033:0x7fbdd6057a99

Add a check for key length in hfs_bnode_read_key to prevent out-of-bounds memory access. If the key length is invalid, the key buffer is cleared, improving stability and reliability.

Fixes: [1da177e4c3f4](#) ("Linux-2.6.12-rc2")

Reported-by: [syzbot+5f3a973ed3dfb85a6683@syzkaller.appspotmail.com](#)

Closes: <https://syzkaller.appspot.com/bug?extid=5f3a973ed3dfb85a6683>

Cc: stable@vger.kernel.org
Signed-off-by: Vasiliy Kovalev <kovalev@altlinux.org>
Link: <https://lore.kernel.org/20241019191303.24048-1-kovalev@altlinux.org>
Reviewed-by: Cengiz Can <cengiz.can@canonical.com>
Signed-off-by: Christian Brauner <brauner@kernel.org>
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

Diffstat

-rw-r--r--	fs/hfs/bnode.c	6
-rw-r--r--	fs/hfsplus/bnode.c	6

2 files changed, 12 insertions, 0 deletions

diff --git a/fs/hfs/bnode.c b/fs/hfs/bnode.c

index 6add6ebfef8967..cb823a8a6ba960 100644

--- a/fs/hfs/bnode.c

+++ b/fs/hfs/bnode.c

```
@@ -67,6 +67,12 @@ void hfs_bnode_read_key(struct hfs_bnode *node, void *key, int off)
     else
         key_len = tree->max_key_len + 1;

+    if (key_len > sizeof(hfs_btree_key) || key_len < 1) {
+        memset(key, 0, sizeof(hfs_btree_key));
+        pr_err("hfs: Invalid key length: %d\n", key_len);
+        return;
+    }
+    hfs_bnode_read(node, key, off, key_len);
}
```

diff --git a/fs/hfsplus/bnode.c b/fs/hfsplus/bnode.c

index 87974d5e679156..079ea80534f7de 100644

--- a/fs/hfsplus/bnode.c

+++ b/fs/hfsplus/bnode.c

```
@@ -67,6 +67,12 @@ void hfs_bnode_read_key(struct hfs_bnode *node, void *key, int off)
     else
         key_len = tree->max_key_len + 2;

+    if (key_len > sizeof(hfsplus_btree_key) || key_len < 1) {
+        memset(key, 0, sizeof(hfsplus_btree_key));
+        pr_err("hfsplus: Invalid key length: %d\n", key_len);
+        return;
+    }
+    hfs_bnode_read(node, key, off, key_len);
}
```