





```

> [<c0176166>] generic_shutdown_super+0x18/0xbe
> [<c017622c>] kill_block_super+0x20/0x32
> [<c01762ce>] deactivate_super+0x3f/0x51
> [<c018724d>] mntput_no_expire+0x42/0x6b
> [<c017a719>] path_release+0x20/0x23
> [<f886387d>] ecryptfs_get_sb+0x45a/0x4ad [ecryptfs]
> [<c0176361>] vfs_kern_mount+0x81/0xf1
> [<c0176419>] do_kern_mount+0x30/0x42
> [<c018847b>] do_mount+0x601/0x678
> [<c0188561>] sys_mount+0x6f/0xa9
> [<c0104f2c>] sysenter_past_esp+0x5d/0x99
> =====
> Code: 30 02 00 00 89 44 24 18 8b 45 ec 89 4c 24 14 89 74 24 10 89 7c 24 0c 89 5c 24 04 89
44 24 08 c7 04 24 ee 9d 4f c0 e8 b7 74 fa ff <0f> 0b eb fe 8b 73 30 39 de 75 04 31 f6 eb 03 f0 ff
0e 8d 43 48
> EIP: [<c01831a7>] shrink_dcache_for_umount_subtree+0x159/0x1fb SS:ESP 0068:f4269c7c
>

```

> This is easy to reproduce just try to mount ecryptfs to nonexisting lower path  
> # mount -tecryptfs private/this\_dir\_not\_exist root -ocipher=aes  
This is fun but ecryptfs\_read\_super() code is more crappy when it looks at first blush.

- 1) After path\_lookup succeed we don't have any guarantee what it is DIR.  
This must be checked explicitly.
- 2) path\_lookup can't return negative dentry, So inode check is useless.

Following patch is updated version and have to be applied instead of first patch  
Signed-off-by: Dmitry Monakhov <dmonakhov@openvz.org>

-----

```

diff --git a/fs/ecryptfs/main.c b/fs/ecryptfs/main.c
index 80044d1..fc4a3a2 100644
--- a/fs/ecryptfs/main.c
+++ b/fs/ecryptfs/main.c
@@ -484,18 +484,12 @@ static int ecryptfs_read_super(struct super_block *sb, const char
*dev_name)
    struct vfsmount *lower_mnt;

    memset(&nd, 0, sizeof(struct nameidata));
- rc = path_lookup(dev_name, LOOKUP_FOLLOW, &nd);
+ rc = path_lookup(dev_name, LOOKUP_FOLLOW | LOOKUP_DIRECTORY, &nd);
    if (rc) {
        ecryptfs_printk(KERN_WARNING, "path_lookup() failed\n");
- goto out_free;
+ goto out;
    }
    lower_root = nd.dentry;
- if (!lower_root->d_inode) {
- ecryptfs_printk(KERN_WARNING,

```

```
- "No directory to interpose on\n");  
- rc = -ENOENT;  
- goto out_free;  
- }  
  lower_mnt = nd.mnt;  
  ecryptfs_set_superblock_lower(sb, lower_root->d_sb);  
  sb->s_maxbytes = lower_root->d_sb->s_maxbytes;
```

---