
Subject: Re: [patch 7/9] unprivileged mounts: allow unprivileged fuse mounts

Posted by [serue](#) on Mon, 14 Jan 2008 23:24:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quoting Miklos Szeredi (miklos@szeredi.hu):

> From: Miklos Szeredi <mszeredi@suse.cz>

>

> Use FS_SAFE for "fuse" fs type, but not for "fuseblk".

>

> FUSE was designed from the beginning to be safe for unprivileged users. This

> has also been verified in practice over many years. In addition unprivileged

> mounts require the parent mount to be owned by the user, which is more strict

> than the current userspace policy.

>

> This will enable future installations to remove the suid-root fusermount

> utility.

>

> Don't require the "user_id=" and "group_id=" options for unprivileged mounts,

> but if they are present, verify them for sanity.

>

> Disallow the "allow_other" option for unprivileged mounts.

>

> Signed-off-by: Miklos Szeredi <mszeredi@suse.cz>

Sounds like a sysctl to enable FS_SAFE for fuse will make this patch acceptable to everyone?

> ---

>

> Index: linux/fs/fuse/inode.c

> =====

> --- linux.orig/fs/fuse/inode.c 2008-01-03 17:13:13.000000000 +0100

> +++ linux/fs/fuse/inode.c 2008-01-03 21:28:01.000000000 +0100

> @@ -357,6 +357,19 @@ static int parse_fuse_opt(char *opt, str

> d->max_read = ~0;

> d->blksize = 512;

>

> + /*

> + * For unprivileged mounts use current uid/gid. Still allow

> + * "user_id" and "group_id" options for compatibility, but

> + * only if they match these values.

> + */

> + if (!capable(CAP_SYS_ADMIN)) {

> + d->user_id = current->uid;

> + d->user_id_present = 1;

> + d->group_id = current->gid;

> + d->group_id_present = 1;

> +

```

> + }
> +
> while ((p = strsep(&opt, ",")) != NULL) {
>     int token;
>     int value;
> @@ -385,6 +398,8 @@ static int parse_fuse_opt(char *opt, str
>     case OPT_USER_ID:
>         if (match_int(&args[0], &value))
>             return 0;
> +     if (d->user_id_present && d->user_id != value)
> +         return 0;
>         d->user_id = value;
>         d->user_id_present = 1;
>         break;
> @@ -392,6 +407,8 @@ static int parse_fuse_opt(char *opt, str
>     case OPT_GROUP_ID:
>         if (match_int(&args[0], &value))
>             return 0;
> +     if (d->group_id_present && d->group_id != value)
> +         return 0;
>         d->group_id = value;
>         d->group_id_present = 1;
>         break;
> @@ -596,6 +613,10 @@ static int fuse_fill_super(struct super_
>     if (!parse_fuse_opt((char *) data, &d, is_bdev))
>         return -EINVAL;
>
> + /* This is a privileged option */
> + if ((d.flags & FUSE_ALLOW_OTHER) && !capable(CAP_SYS_ADMIN))
> +     return -EPERM;
> +
>     if (is_bdev) {
>         #ifdef CONFIG_BLOCK
>         if (!sb_set_blocksize(sb, d.blksize))
> @@ -696,9 +717,9 @@ static int fuse_get_sb(struct file_syste
>     static struct file_system_type fuse_fs_type = {
>         .owner = THIS_MODULE,
>         .name = "fuse",
>         .fs_flags = FS_HAS_SUBTYPE,
>         .get_sb = fuse_get_sb,
>         .kill_sb = kill_anon_super,
> +     .fs_flags = FS_HAS_SUBTYPE | FS_SAFE,
>     };
>
>     #ifdef CONFIG_BLOCK
>
> --

```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
