xorl %eax, %eax

CVE-2013-1848: Linux kernel EXT3 ext3_msg() Format String

leave a comment »

Recently Lars-Peter Clausen committed a change on Linux kernel that fixes a format string vulnerability in the EXT3 filesystem code. The susceptible code resides in fs/ext3/super.c but to better understand it we need to have a look on how ext3_msg() is defined first.

```
void ext3_msg(struct super_block *sb, const char *prefix,
                const char *fmt, ...)
{
        struct va_format vaf;
        va_list args;

        va_start(args, fmt);

        vaf.fmt = fmt;
        vaf.va = &args;

        printk("%sEXT3-fs (%s): %pV\n", prefix, sb->s_id, &vaf);

        va_end(args);
}
```

So, it should be called passing the following three mandatory arguments:
– Pointer to the super-block structure
– Prefix string
– Format string
And of course, any variables to be printed. As Lars-Peter Clausen noticed, there were two cases where there was no prefix defined. This makes the format string argument to be passed as prefix and any variables to be processed as the format string. Here are these two cases:

```
/*
 * Open the external journal device
 */
static struct block_device *ext3_blkdev_get(dev_t dev, struct super_block *sb
{
  ...
fail:
        ext3_msg(sb, "error: failed to open journal device %s: %ld",
```

And…

```
static ext3_fsblk_t get_sb_block(void **data, struct super_block *sb)
{
        ext3_fsblk_t    sb_block;
  ...
        if (*options && *options != ',') {
                ext3_msg(sb, "error: invalid sb specification: %s",
                       (char *) *data);
  ...
        return sb_block;
}
```

The fix was to add the missing prefix argument to the function call like this.

```
@@ -353,7 +353,7 @@ static struct block_device *ext3_blkdev_get(dev_t dev, str
    return bdev;
 fail:
-   ext3_msg(sb, "error: failed to open journal device %s: %ld",
+   ext3_msg(sb, KERN_ERR, "error: failed to open journal device %s: %ld",
        __bdevname(dev, b), PTR_ERR(bdev));
    return NULL;
@@ -887,7 +887,7 @@ static ext3_fsblk_t get_sb_block(void **data, struct super
    /*todo: use simple_strtoll with >32bit ext3 */
    sb_block = simple_strtoul(options, &options, 0);
    if (*options && *options != ',') {
-       ext3_msg(sb, "error: invalid sb specification: %s",
+       ext3_msg(sb, KERN_ERR, "error: invalid sb specification: %s",
               (char *) *data);
        return 1;
    }
```

Written by xorl

May 21, 2013 at 21:15

Posted in linux, vulnerabilities