Subject: [PATCH 6/6] userns strict: hook ext3
Posted by serue on Mon, 04 Jun 2007 19:42:20 GMT
View Forum Message <> Reply to Message

>From nobody Mon Sep 17 00:00:00 2001
From: Serge Hallyn <serue@us.ibm.com>
Date: Wed, 28 Mar 2007 13:11:19 -0500
Subject: [PATCH 6/6] userns strict: hook ext3

Add a user namespace pointer to the ext3 superblock and inode.

Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>

```
 fs/ext3/acl.c             |    4 +++-
 fs/ext3/balloc.c          |   13 ++++++++++---
 fs/ext3/ialloc.c          |    5 +++++
 fs/ext3/inode.c           |    2 ++
 fs/ext3/ioctl.c           |   18 ++++++++++++-------
 fs/ext3/super.c           |    4 ++++
 fs/ext3/xattr_trusted.c   |    3 ++-
 include/linux/ext3_fs_sb.h |    1 +
 8 files changed, 38 insertions(+), 12 deletions(-)

d51e324bced6f12d2fc7906dbfc355f4cd2bc8bc
diff --git a/fs/ext3/acl.c b/fs/ext3/acl.c
index 1e5038d..35b64a9 100644
--- a/fs/ext3/acl.c
+++ b/fs/ext3/acl.c
@@ -11,6 +11,7 @@ #include <linux/capability.h>
 #include <linux/fs.h>
 #include <linux/ext3_jbd.h>
 #include <linux/ext3_fs.h>
+#include <linux/user_namespace.h>
 #include "xattr.h"
 #include "acl.h"

@@ -489,7 +490,8 @@ ext3_xattr_set_acl(struct inode *inode,

  if (!test_opt(inode->i_sb, POSIX_ACL))
   return -EOPNOTSUPP;
- if ((current->fsuid != inode->i_uid) && !capable(CAP_FOWNER))
+ if (!task_inode_same_fsuid(current, inode) &&
+   !task_ino_capable(inode, CAP_FOWNER))
   return -EPERM;

  if (value) {
```

```
diff --git a/fs/ext3/balloc.c b/fs/ext3/balloc.c
index ca8aee6..ab4076d 100644
--- a/fs/ext3/balloc.c
+++ b/fs/ext3/balloc.c
@@ -19,6 +19,7 @@ #include <linux/ext3_fs.h>
 #include <linux/ext3_jbd.h>
 #include <linux/quotaops.h>
 #include <linux/buffer_head.h>
+#include <linux/user_namespace.h>

 /*
  * balloc.c contains the blocks allocation and deallocation routines
@@ -1359,9 +1360,15 @@ static int ext3_has_free_blocks(struct e

  free_blocks = percpu_counter_read_positive(&sbi->s_freeblocks_counter);
  root_blocks = le32_to_cpu(sbi->s_es->s_r_blocks_count);
- if (free_blocks < root_blocks + 1 && !capable(CAP_SYS_RESOURCE) &&
-  sbi->s_resuid != current->fsuid &&
-  (sbi->s_resgid == 0 || !in_group_p (sbi->s_resgid))) {
+ if (free_blocks < root_blocks + 1) {
+  if (sbi->s_resuidns != task_user_ns(current))
+   return 0;
+  if (capable(CAP_SYS_RESOURCE))
+   return 1;
+  if (sbi->s_resuid == current->fsuid)
+   return 1;
+  if (sbi->s_resgid != 0 && in_group_p (sbi->s_resgid))
+   return 1;
   return 0;
  }
  return 1;
diff --git a/fs/ext3/ialloc.c b/fs/ext3/ialloc.c
index e45dbd6..eb31b83 100644
--- a/fs/ext3/ialloc.c
+++ b/fs/ext3/ialloc.c
@@ -23,6 +23,7 @@ #include <linux/quotaops.h>
 #include <linux/buffer_head.h>
 #include <linux/random.h>
 #include <linux/bitops.h>
+#include <linux/user_namespace.h>

 #include <asm/byteorder.h>

@@ -133,6 +134,9 @@ void ext3_free_inode (handle_t *handle,
  /* Do this BEFORE marking the inode not in use or returning an error */
  clear_inode (inode);

+ put_user_ns(inode->i_userns);
```

```
+ inode->i_userns = NULL;
+
  es = EXT3_SB(sb)->s_es;
  if (ino < EXT3_FIRST_INO(sb) || ino > le32_to_cpu(es->s_inodes_count)) {
   ext3_error (sb, "ext3_free_inode",
@@ -547,6 +551,7 @@ got:
  sb->s_dirt = 1;

  inode->i_uid = current->fsuid;
+ inode->i_userns = get_task_user_ns(current);
  if (test_opt (sb, GRPID))
   inode->i_gid = dir->i_gid;
  else if (dir->i_mode & S_ISGID) {
diff --git a/fs/ext3/inode.c b/fs/ext3/inode.c
index 535c5a5..2ec63bb 100644
--- a/fs/ext3/inode.c
+++ b/fs/ext3/inode.c
@@ -37,6 +37,7 @@ #include <linux/writeback.h>
 #include <linux/mpage.h>
 #include <linux/uio.h>
 #include <linux/bio.h>
+#include <linux/user_namespace.h>
 #include "xattr.h"
 #include "acl.h"

@@ -2667,6 +2668,7 @@ #endif
  inode->i_mode = le16_to_cpu(raw_inode->i_mode);
  inode->i_uid = (uid_t)le16_to_cpu(raw_inode->i_uid_low);
  inode->i_gid = (gid_t)le16_to_cpu(raw_inode->i_gid_low);
+ inode->i_userns = get_task_user_ns(current);
  if(!(test_opt (inode->i_sb, NO_UID32))) {
   inode->i_uid |= le16_to_cpu(raw_inode->i_uid_high) << 16;
   inode->i_gid |= le16_to_cpu(raw_inode->i_gid_high) << 16;
diff --git a/fs/ext3/ioctl.c b/fs/ext3/ioctl.c
index 9b8090d..6874367 100644
--- a/fs/ext3/ioctl.c
+++ b/fs/ext3/ioctl.c
@@ -15,6 +15,7 @@ #include <linux/ext3_jbd.h>
 #include <linux/time.h>
 #include <linux/compat.h>
 #include <linux/smp_lock.h>
+#include <linux/user_namespace.h>
 #include <asm/uaccess.h>

 int ext3_ioctl (struct inode * inode, struct file * filp, unsigned int cmd,
@@ -40,7 +41,8 @@ int ext3_ioctl (struct inode * inode, st
   if (IS_RDONLY(inode))
    return -EROFS;
```

```
-  if ((current->fsuid != inode->i_uid) && !capable(CAP_FOWNER))
+   if (!task_inode_same_fsuid(current, inode) &&
+    !task_ino_capable(inode, CAP_FOWNER))
    return -EACCES;

   if (get_user(flags, (int __user *) arg))
@@ -62,7 +64,7 @@ int ext3_ioctl (struct inode * inode, st
    * This test looks nicer. Thanks to Pauline Middelink
    */
   if ((flags ^ oldflags) & (EXT3_APPEND_FL | EXT3_IMMUTABLE_FL)) {
-   if (!capable(CAP_LINUX_IMMUTABLE)) {
+   if (!task_ino_capable(inode, CAP_LINUX_IMMUTABLE)) {
     mutex_unlock(&inode->i_mutex);
     return -EPERM;
    }
@@ -73,7 +75,7 @@ int ext3_ioctl (struct inode * inode, st
    * the relevant capability.
    */
   if ((jflag ^ oldflags) & (EXT3_JOURNAL_DATA_FL)) {
-   if (!capable(CAP_SYS_RESOURCE)) {
+   if (!task_ino_capable(inode, CAP_SYS_RESOURCE)) {
     mutex_unlock(&inode->i_mutex);
     return -EPERM;
    }
@@ -121,7 +123,8 @@ flags_err:
   __u32 generation;
   int err;

-  if ((current->fsuid != inode->i_uid) && !capable(CAP_FOWNER))
+  if (!task_inode_same_fsuid(current, inode) &&
+    !task_ino_capable(inode, CAP_FOWNER))
    return -EPERM;
   if (IS_RDONLY(inode))
    return -EROFS;
@@ -180,7 +183,8 @@ #endif
   if (IS_RDONLY(inode))
    return -EROFS;

-  if ((current->fsuid != inode->i_uid) && !capable(CAP_FOWNER))
+  if (!task_inode_same_fsuid(current, inode) &&
+    !task_ino_capable(inode, CAP_FOWNER))
    return -EACCES;

   if (get_user(rsv_window_size, (int __user *)arg))
@@ -209,7 +213,7 @@ #endif
   struct super_block *sb = inode->i_sb;
   int err;
```

```
-  if (!capable(CAP_SYS_RESOURCE))
+  if (!task_ino_capable(inode, CAP_SYS_RESOURCE))
     return -EPERM;

   if (IS_RDONLY(inode))
@@ -230,7 +234,7 @@ #endif
   struct super_block *sb = inode->i_sb;
   int err;

-  if (!capable(CAP_SYS_RESOURCE))
+  if (!task_ino_capable(inode, CAP_SYS_RESOURCE))
     return -EPERM;

   if (IS_RDONLY(inode))
diff --git a/fs/ext3/super.c b/fs/ext3/super.c
index 80dfa27..19e32ee 100644
--- a/fs/ext3/super.c
+++ b/fs/ext3/super.c
@@ -35,6 +35,7 @@ #include <linux/mount.h>
 #include <linux/namei.h>
 #include <linux/quotaops.h>
 #include <linux/seq_file.h>
+#include <linux/user_namespace.h>

 #include <asm/uaccess.h>

@@ -403,6 +404,7 @@ static void ext3_put_super (struct super
  for (i = 0; i < sbi->s_gdb_count; i++)
   brelse(sbi->s_group_desc[i]);
  kfree(sbi->s_group_desc);
+ put_user_ns(sbi->s_resuidns);
  percpu_counter_destroy(&sbi->s_freeblocks_counter);
  percpu_counter_destroy(&sbi->s_freeinodes_counter);
  percpu_counter_destroy(&sbi->s_dirs_counter);
@@ -1417,6 +1419,7 @@ static int ext3_fill_super (struct super
  sbi->s_mount_opt = 0;
  sbi->s_resuid = EXT3_DEF_RESUID;
  sbi->s_resgid = EXT3_DEF_RESGID;
+ sbi->s_resuidns = get_task_user_ns(current);

  unlock_kernel();

@@ -1825,6 +1828,7 @@ #endif
  brelse(bh);
 out_fail:
  sb->s_fs_info = NULL;
+ put_user_ns(sbi->s_resuidns);
```

```
  kfree(sbi);
  lock_kernel();
  return -EINVAL;
diff --git a/fs/ext3/xattr_trusted.c b/fs/ext3/xattr_trusted.c
index 86d91f1..d37bfdd 100644
--- a/fs/ext3/xattr_trusted.c
+++ b/fs/ext3/xattr_trusted.c
@@ -12,6 +12,7 @@ #include <linux/fs.h>
 #include <linux/smp_lock.h>
 #include <linux/ext3_jbd.h>
 #include <linux/ext3_fs.h>
+#include <linux/user_namespace.h>
 #include "xattr.h"

 #define XATTR_TRUSTED_PREFIX "trusted."
@@ -23,7 +24,7 @@ ext3_xattr_trusted_list(struct inode *in
  const size_t prefix_len = sizeof(XATTR_TRUSTED_PREFIX)-1;
  const size_t total_len = prefix_len + name_len + 1;

- if (!capable(CAP_SYS_ADMIN))
+ if (!task_ino_capable(inode, CAP_SYS_ADMIN))
   return 0;

  if (list && total_len <= list_size) {
diff --git a/include/linux/ext3_fs_sb.h b/include/linux/ext3_fs_sb.h
index f61309c..2a68a21 100644
--- a/include/linux/ext3_fs_sb.h
+++ b/include/linux/ext3_fs_sb.h
@@ -44,6 +44,7 @@ struct ext3_sb_info {
  unsigned long  s_mount_opt;
  uid_t s_resuid;
  gid_t s_resgid;
+ struct user_namespace *s_resuidns;
  unsigned short s_mount_state;
  unsigned short s_pad;
  int s_addr_per_block_bits;
--
1.3.2
```