

[patch 05/14] remap_file_pages protection support: cleanup syscall checks

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-04/msg06743.html>

- *From:* blaisorblade@xxxxxxxx
 - *Date:* Sun, 30 Apr 2006 19:29:58 +0200
-

From: Paolo 'Blaisorblade' Giarrusso <blaisorblade@xxxxxxxx>

This patch reorganizes the code only, without differences in behaviour. It makes the code more readable on its own, and is needed for next patches. I've split this out to avoid cluttering real patches.

*) remap_file_pages protection support: use EOVERFLOW ret code

Use -EOVERFLOW ("Value too large for defined data type") rather than -EINVAL when we cannot store the file offset in the PTE.

Signed-off-by: Paolo 'Blaisorblade' Giarrusso <blaisorblade@xxxxxxxx>

Index: linux-2.6.git/mm/fremap.c

```
=====
--- linux-2.6.git.orig/mm/fremap.c
+++ linux-2.6.git/mm/fremap.c
@@ -140,7 +140,7 @@ out:
 * future.
 */
asmlinkage long sys_remap_file_pages(unsigned long start, unsigned long size,
- unsigned long __prot, unsigned long pgoff, unsigned long flags)
+ unsigned long prot, unsigned long pgoff, unsigned long flags)
{
struct mm_struct *mm = current->mm;
struct address_space *mapping;
@@ -148,9 +148,10 @@ asmlinkage long sys_remap_file_pages(uns
struct vm_area_struct *vma;
int err = -EINVAL;
int has_write_lock = 0;
+ pgprot_t pgprot;

- if (__prot)
- return err;
+ if (prot)
+ goto out;
/*
 * Sanitize the syscall parameters:
 */
```

[patch 05/14] remap_file_pages protection support: cleanup syscall checks

```
@@ -159,17 +160,19 @@ asmlinkage long sys_remap_file_pages(uns
```

```
/* Does the address range wrap, or is the span zero-sized? */  
if (start + size <= start)  
- return err;  
+ goto out;  
  
/* Can we represent this offset inside this architecture's pte's? */  
#if PTE_FILE_MAX_BITS < BITS_PER_LONG  
- if (pgoff + (size >> PAGE_SHIFT) >= (1UL << PTE_FILE_MAX_BITS))  
- return err;  
+ if (pgoff + (size >> PAGE_SHIFT) >= (1UL << PTE_FILE_MAX_BITS)) {  
+ err = -E_OVERFLOW;  
+ goto out;  
+ }  
#endif  
  
/* We need down_write() to change vma->vm_flags. */  
down_read(&mm->mmap_sem);  
- retry:  
+retry:  
vma = find_vma(mm, start);
```

```
/*  
@@ -178,12 +181,21 @@ asmlinkage long sys_remap_file_pages(uns  
* the single existing vma. vm_private_data is used as a  
* swapout cursor in a VM_NONLINEAR vma.  
*/  
- if (vma && (vma->vm_flags & VM_SHARED) &&  
- (!vma->vm_private_data || (vma->vm_flags & VM_NONLINEAR)) &&  
- vma->vm_ops && vma->vm_ops->populate &&  
- end > start && start >= vma->vm_start &&  
- end <= vma->vm_end) {  
+ if (!vma)  
+ goto out_unlock;  
+  
+ if (!(vma->vm_flags & VM_SHARED))  
+ goto out_unlock;  
+  
+ if (!vma->vm_ops || !vma->vm_ops->populate)  
+ goto out_unlock;  
  
+ if (end <= start || start < vma->vm_start || end > vma->vm_end)  
+ goto out_unlock;  
+  
+ pgprot = vma->vm_page_prot;  
+  
+ if (!vma->vm_private_data || (vma->vm_flags & VM_NONLINEAR)) {  
/* Must set VM_NONLINEAR before any pages are populated. */  
if (pgoff != linear_page_index(vma, start) &&  
!(vma->vm_flags & VM_NONLINEAR)) {
```

[patch 05/14] remap_file_pages protection support: cleanup syscall checks

[patch 05/14] remap_file_pages protection support: cleanup syscall checks

```
@@ -203,9 +215,8 @@ asmlinkage long sys_remap_file_pages(uns
spin_unlock(&mapping->i_mmap_lock);
}

- err = vma->vm_ops->populate(vma, start, size,
- vma->vm_page_prot,
- pgoff, flags & MAP_NONBLOCK);
+ err = vma->vm_ops->populate(vma, start, size, pgprot, pgoff,
+ flags & MAP_NONBLOCK);

/*
 * We would like to clear VM_NONLINEAR, in the case when
@@ -214,11 +225,14 @@ asmlinkage long sys_remap_file_pages(uns
 * successful populate, and have no way to upgrade sem.
 */
}
+
+out_unlock:
if (likely(!has_write_lock))
up_read(&mm->mmap_sem);
else
up_write(&mm->mmap_sem);

+out:
return err;
}
```

--
Inform me of my mistakes, so I can keep imitating Homer Simpson's "Doh!".
Paolo Giarrusso, aka Blaisorblade (Skype ID "PaoloGiarrusso", ICQ 215621894)
<http://www.user-mode-linux.org/~blaisorblade>

-
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>