

[PATCH 2/9] x86-64 untangle seccomp.h vs thread_info

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-11/9629.html>

From: Benjamin LaHaise (*bcr1_at_kvack.org*)

Date: 11/30/05

Date: Tue, 29 Nov 2005 23:21:31 -0500
To: Andi Kleen <ak@suse.de>

When compiling on x86-64 SMP, seccomp.h introduces an ordering dependency on asm/thread_info.h by using inline functions which reference the thread info declaration before struct task_struct is finished being defined. In order to avoid this nasty include mess, convert the definitions in seccomp.h into macros.

```
---
include/linux/seccomp.h | 15 ++++++-----
1 files changed, 6 insertions(+), 9 deletions(-)
applies-to: 610619c4af824b70bd81f228b24357838223ee50
194f04991a44fec3b21d30f1a137d402132996fc
diff --git a/include/linux/seccomp.h b/include/linux/seccomp.h
index dc89116..61eabc3 100644
--- a/include/linux/seccomp.h
+++ b/include/linux/seccomp.h
@@ -13,16 +13,13 @@
 typedef struct { int mode; } seccomp_t;

extern void __secure_computing(int);
-static inline void secure_computing(int this_syscall)
-{
-    if (unlikely(test_thread_flag(TIF_SECCOMP)))
-        __secure_computing(this_syscall);
-}
+#define secure_computing(this_syscall) \
+do { \
+    if (unlikely(test_thread_flag(TIF_SECCOMP))) \
+        __secure_computing(this_syscall); \
+} while (0)

-static inline int has_secure_computing(struct thread_info *ti)
-{
-    return unlikely(test_ti_thread_flag(ti, TIF_SECCOMP));
-}
+#define has_secure_computing(ti) unlikely(test_ti_thread_flag(ti, TIF_SECCOMP))

#else /* CONFIG_SECCOMP */
```

0.99.9.GIT
-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in

Linux-Kernel: [PATCH 2/9] x86-64 untangle seccomp.h vs thread_info

the body of a message to majordomo@vger.kernel.org

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>