author	Russell King <rmk+kernel@arm.linux.org.uk>	2012-09-07 17:22:28 (GMT)
committer	Gerrit - the friendly Code Review server <code-review@localhost>	2013-07-15 22:50:58 (GMT)
commit	76565e3d786bed66f247c682bd9f591098522483 (patch)	
tree	125f1729702ecd85f376cef4d2bd58a74998d5e8	
parent	52c75b5feccc5fcf45e24786606c4d6868249b6f (diff)	

diff options

context:	3 ▼
space:	include ▼
mode:	unified ▼

ARM: 7527/1: uaccess: explicitly check __user pointer when !CPU_USE_DOMAINS

```
The {get,put}_user macros don't perform range checking on the provided
__user address when !CPU_HAS_DOMAINS.

This patch reworks the out-of-line assembly accessors to check the user
address against a specified limit, returning -EFAULT if is is out of
range.

[will: changed get_user register allocation to match put_user]
[rmk: fixed building on older ARM architectures]

CRs-Fixed: 504011
Change-Id: I3818045a136fcdf72deb1371b132e090fd7ed643
Reported-by: Catalin Marinas <catalin.marinas@arm.com>
Signed-off-by: Will Deacon <will.deacon@arm.com>
Cc: stable@vger.kernel.org
Signed-off-by: Russell King <rmk+kernel@arm.linux.org.uk>
Git-commit: 8404663f81d212918ff85f493649a7991209fa04
Git-repo: git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git
Signed-off-by: Laura Abbott <lauraa@codeaurora.org>
```

Diffstat

-rw-r--r--	arch/arm/include/asm/assembler.h	8
-rw-r--r--	arch/arm/include/asm/uaccess.h	40
-rw-r--r--	arch/arm/lib/getuser.S	23
-rw-r--r--	arch/arm/lib/putuser.S	6

4 files changed, 56 insertions, 21 deletions

```
diff --git a/arch/arm/include/asm/assembler.h b/arch/arm/include/asm/assembler.h
index 03fb936..5c8b3bf4 100644
--- a/arch/arm/include/asm/assembler.h
+++ b/arch/arm/include/asm/assembler.h
@@ -320,4 +320,12 @@
        .size \name , . - \name
        .endm

+       .macro check_uaccess, addr:req, size:req, limit:req, tmp:req, bad:req
+#ifndef CONFIG_CPU_USE_DOMAINS
+       adds    \tmp, \addr, #\size - 1
+       sbcccs  \tmp, \tmp, \limit
+       bcs     \bad
+#endif
+       .endm
+
 #endif /* __ASM_ASSEMBLER_H__ */

diff --git a/arch/arm/include/asm/uaccess.h b/arch/arm/include/asm/uaccess.h
index 71f6536..0a070e9 100644
--- a/arch/arm/include/asm/uaccess.h
+++ b/arch/arm/include/asm/uaccess.h
@@ -101,28 +101,39 @@ extern int __get_user_1(void *);
 extern int __get_user_2(void *);
 extern int __get_user_4(void *);

-#define __get_user_x(__r2,__p,__e,__s,__i...)                          \
+#define __GUP_CLOBBER_1        "lr", "cc"
+#ifdef CONFIG_CPU_USE_DOMAINS
+#define __GUP_CLOBBER_2        "ip", "lr", "cc"
+#else
+#define __GUP_CLOBBER_2 "lr", "cc"
+#endif
+#define __GUP_CLOBBER_4        "lr", "cc"
+
+#define __get_user_x(__r2,__p,__e,__l,__s)                             \
           __asm__ __volatile__ (                                       \
                __asmeq("%0", "r0") __asmeq("%1", "r2")                 \
+              __asmeq("%3", "r1")                                     \
               "bl     __get_user_" #__s                               \
               : "=&r" (__e), "=r" (__r2)                              \
-              : "0" (__p)                                             \
-              : __i, "cc")
+              : "0" (__p), "r" (__l)                                  \
+              : __GUP_CLOBBER_##__s)

 #define get_user(x,p)                                                  \
        ({                                                              \
+              unsigned long __limit = current_thread_info()->addr_limit - 1; \
               register const typeof(*(p)) __user *__p asm("r0") = (p);\
               register unsigned long __r2 asm("r2");                  \
+              register unsigned long __l asm("r1") = __limit;         \
               register int __e asm("r0");                             \
               switch (sizeof(*(__p))) {                               \
                case 1:                                                  \
-                       __get_user_x(__r2, __p, __e, 1, "lr");           \
-                       break;                                           \
+                       __get_user_x(__r2, __p, __e, __l, 1);            \
+                       break;                                           \
                case 2:                                                  \
-                       __get_user_x(__r2, __p, __e, 2, "r3", "lr");     \
+                       __get_user_x(__r2, __p, __e, __l, 2);            \
                        break;                                           \
                case 4:                                                  \
-                       __get_user_x(__r2, __p, __e, 4, "lr");           \
+                       __get_user_x(__r2, __p, __e, __l, 4);            \
                        break;                                           \
                default: __e = __get_user_bad(); break;                  \
                }                                                        \
@@ -135,31 +146,34 @@ extern int __put_user_2(void *, unsigned int);
 extern int __put_user_4(void *, unsigned int);
 extern int __put_user_8(void *, unsigned long long);
 
-#define __put_user_x(__r2,__p,__e,__s)                                  \
+#define __put_user_x(__r2,__p,__e,__l,__s)                              \
           __asm__ __volatile__ (                                        \
                __asmeq("%0", "r0") __asmeq("%2", "r2")                  \
+               __asmeq("%3", "r1")                                      \
                "bl     __put_user_" #__s                                \
                : "=&r" (__e)                                            \
-               : "0" (__p), "r" (__r2)                                  \
+               : "0" (__p), "r" (__r2), "r" (__l)                       \
               : "ip", "lr", "cc")

 #define put_user(x,p)                                                   \
        ({                                                               \
+               unsigned long __limit = current_thread_info()->addr_limit - 1; \
               register const typeof(*(p)) __r2 asm("r2") = (x);        \
               register const typeof(*(p)) __user *__p asm("r0") = (p);\
+               register unsigned long __l asm("r1") = __limit;          \
               register int __e asm("r0");                              \
               switch (sizeof(*(__p))) {                                \
               case 1:                                                  \
-                       __put_user_x(__r2, __p, __e, 1);                 \
+                       __put_user_x(__r2, __p, __e, __l, 1);            \
                       break;                                           \
               case 2:                                                  \
-                       __put_user_x(__r2, __p, __e, 2);                 \
+                       __put_user_x(__r2, __p, __e, __l, 2);            \
                       break;                                           \
               case 4:                                                  \
-                       __put_user_x(__r2, __p, __e, 4);                 \
+                       __put_user_x(__r2, __p, __e, __l, 4);            \
                       break;                                           \
               case 8:                                                  \
-                       __put_user_x(__r2, __p, __e, 8);                 \
+                       __put_user_x(__r2, __p, __e, __l, 8);            \
                       break;                                           \
               default: __e = __put_user_bad(); break;                  \
               }                                                        \

diff --git a/arch/arm/lib/getuser.S b/arch/arm/lib/getuser.S
index 11093a7..9b06bb4 100644
--- a/arch/arm/lib/getuser.S
+++ b/arch/arm/lib/getuser.S
@@ -16,8 +16,9 @@
  * __get_user_X
  *
  * Inputs:     r0 contains the address
+ *             r1 contains the address limit, which must be preserved
  * Outputs:    r0 is the error code
- *             r2, r3 contains the zero-extended value
+ *             r2 contains the zero-extended value
  *             lr corrupted
  *
  * No other registers must be altered.  (see <asm/uaccess.h>
@@ -27,33 +28,39 @@
  * Note also that it is intended that __get_user_bad is not global.
  */
 #include <linux/linkage.h>
+#include <asm/assembler.h>
 #include <asm/errno.h>
 #include <asm/domain.h>

 ENTRY(__get_user_1)
+       check_uaccess r0, 1, r1, r2, __get_user_bad
 1: TUSER(ldrb) r2, [r0]
        mov     r0, #0
        mov     pc, lr
 ENDPROC(__get_user_1)

 ENTRY(__get_user_2)
-#ifdef CONFIG_THUMB2_KERNEL
-2: TUSER(ldrb) r2, [r0]
-3: TUSER(ldrb) r3, [r0, #1]
+       check_uaccess r0, 2, r1, r2, __get_user_bad
+#ifdef CONFIG_CPU_USE_DOMAINS
+rb     .req    ip
+2:     ldrbt   r2, [r0], #1
+3:     ldrbt   rb, [r0], #0
 #else
-2: TUSER(ldrb) r2, [r0], #1
-3: TUSER(ldrb) r3, [r0]
+rb     .req    r0
+2:     ldrb    r2, [r0]
+3:     ldrb    rb, [r0, #1]
 #endif
 #ifndef __ARMEB__
```

```
-       orr     r2, r2, r3, lsl #8
+       orr     r2, r2, rb, lsl #8
 #else
-       orr     r2, r3, r2, lsl #8
+       orr     r2, rb, r2, lsl #8
 #endif
        mov     r0, #0
        mov     pc, lr
 ENDPROC(__get_user_2)

 ENTRY(__get_user_4)
+       check_uaccess r0, 4, r1, r2, __get_user_bad
 4: TUSER(ldr)  r2, [r0]
        mov     r0, #0
        mov     pc, lr

diff --git a/arch/arm/lib/putuser.S b/arch/arm/lib/putuser.S
index 7db2599..3d73dcb 100644
--- a/arch/arm/lib/putuser.S
+++ b/arch/arm/lib/putuser.S
@@ -16,6 +16,7 @@
  * __put_user_X
  *
  * Inputs:     r0 contains the address
+ *             r1 contains the address limit, which must be preserved
  *             r2, r3 contains the value
  * Outputs:    r0 is the error code
  *             lr corrupted
@@ -27,16 +28,19 @@
  * Note also that it is intended that __put_user_bad is not global.
  */
 #include <linux/linkage.h>
+#include <asm/assembler.h>
 #include <asm/errno.h>
 #include <asm/domain.h>

 ENTRY(__put_user_1)
+       check_uaccess r0, 1, r1, ip, __put_user_bad
 1: TUSER(strb) r2, [r0]
        mov     r0, #0
        mov     pc, lr
 ENDPROC(__put_user_1)

 ENTRY(__put_user_2)
+       check_uaccess r0, 2, r1, ip, __put_user_bad
        mov     ip, r2, lsr #8
 #ifdef CONFIG_THUMB2_KERNEL
 #ifndef __ARMEB__
@@ -60,12 +64,14 @@ ENTRY(__put_user_2)
 ENDPROC(__put_user_2)

 ENTRY(__put_user_4)
+       check_uaccess r0, 4, r1, ip, __put_user_bad
 4: TUSER(str)  r2, [r0]
        mov     r0, #0
        mov     pc, lr
 ENDPROC(__put_user_4)

 ENTRY(__put_user_8)
+       check_uaccess r0, 8, r1, ip, __put_user_bad
 #ifdef CONFIG_THUMB2_KERNEL
 5: TUSER(str)  r2, [r0]
 6: TUSER(str)  r3, [r0, #4]
```

generated by cgit v0.10.2 at 2015-09-27 15:50:55 (GMT)