Subject: [PATCH 1/5] Simplify module_get_kallsym() by dropping length arg
Posted by Alexey Dobriyan on Mon, 02 Apr 2007 14:53:26 GMT
View Forum Message <> Reply to Message

Hi, this is an attempt to fix some races between rmmod and some of those
pesky proc files. rmmod vs wchan race and rmmod vs kallsyms race were
actually triggered.

For the record, initial attempts to plug these races are here
 http://marc.info/?l=linux-kernel&m=117404513602668&w =2
 http://marc.info/?l=linux-kernel&m=117404513702680&w =2
They were done by making module_mutex global (but, of course, not exported)

Now that respin is done I don't know which series I like more. :-\
Probaly, some duplication among address resolution appeared.

Patch #1 and #3 only technically depend on others.

Please, review.
 ---------- ----------
[PATCH 1/5] Simplify module_get_kallsym() by dropping length arg

module_get_kallsym() could in theory truncate module symbol name to fit
in buffer, but nobody does this. Always use KSYM_NAME_LEN + 1 bytes for name.

Suggested by lg^WRusty.

Signed-off-by: Alexey Dobriyan <adobriyan@sw.ru>
---

 include/linux/module.h | 5 ++---
 kernel/kallsyms.c | 2 +-
 kernel/module.c | 5 +++--
 3 files changed, 6 insertions(+), 6 deletions(-)

--- a/include/linux/module.h
+++ b/include/linux/module.h
@@ -373,7 +373,7 @@ int is_module_address(unsigned long addr
 /* Returns module and fills in value, defined and namebuf, or NULL if
    symnum out of range. */
 struct module *module_get_kallsym(unsigned int symnum, unsigned long *value,
-    char *type, char *name, size_t namelen);
+   char *type, char *name);

 /* Look for this name: can be of form module:name. */
 unsigned long module_kallsyms_lookup_name(const char *name);
@@ -529,8 +529,7 @@ static inline const char *module_address

```
 static inline struct module *module_get_kallsym(unsigned int symnum,
        unsigned long *value,
-       char *type, char *name,
-       size_t namelen)
+       char *type, char *name)
 {
  return NULL;
 }
--- a/kernel/kallsyms.c
+++ b/kernel/kallsyms.c
@@ -306,7 +306,7 @@ static int get_ksymbol_mod(struct kallsy
 {
  iter->owner = module_get_kallsym(iter->pos - kallsyms_num_syms,
        &iter->value, &iter->type,
-       iter->name, sizeof(iter->name));
+       iter->name);
  if (iter->owner == NULL)
   return 0;

--- a/kernel/module.c
+++ b/kernel/module.c
@@ -19,6 +19,7 @@
 #include <linux/module.h>
 #include <linux/moduleloader.h>
 #include <linux/init.h>
+#include <linux/kallsyms.h>
 #include <linux/kernel.h>
 #include <linux/slab.h>
 #include <linux/vmalloc.h>
@@ -2120,7 +2121,7 @@ const char *module_address_lookup(unsign
 }

 struct module *module_get_kallsym(unsigned int symnum, unsigned long *value,
-    char *type, char *name, size_t namelen)
+    char *type, char *name)
 {
  struct module *mod;

@@ -2130,7 +2131,7 @@ struct module *module_get_kallsym(unsign
    *value = mod->symtab[symnum].st_value;
    *type = mod->symtab[symnum].st_info;
    strlcpy(name, mod->strtab + mod->symtab[symnum].st_name,
-    namelen);
+    KSYM_NAME_LEN + 1);
    mutex_unlock(&module_mutex);
    return mod;
   }
```