
Subject: [PATCH 2/4] sysctl: Implement sysctl_uts_string
Posted by [ebiederm](#) on Mon, 27 Nov 2006 05:05:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

The problem: When using sys_sysctl we don't read the proper values for the variables exported from the uts namespace, nor do we do the proper locking.

This patch introduces sysctl_uts_string which properly fetches the values and does the proper locking.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

kernel/sysctl.c | 37 ++++++-----
1 files changed, 32 insertions(+), 5 deletions(-)

diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index 0521884..63db5a5 100644

--- a/kernel/sysctl.c

+++ b/kernel/sysctl.c

@@ -136,6 +136,10 @@ #endif

static int proc_do_uts_string(ctl_table *table, int write, struct file *filp,
void __user *buffer, size_t *lenp, loff_t *ppos);

+static int sysctl_uts_string(ctl_table *table, int __user *name, int nlen,

+ void __user *oldval, size_t __user *oldlenp,

+ void __user *newval, size_t newlen, void **context);

+

#ifdef CONFIG_PROC_SYSCTL

static int proc_do_cad_pid(ctl_table *table, int write, struct file *filp,
void __user *buffer, size_t *lenp, loff_t *ppos);

@@ -257,7 +261,7 @@ static ctl_table kern_table[] = {

.maxlen = sizeof(init_uts_ns.name.sysname),

.mode = 0444,

.proc_handler = &proc_do_uts_string,

- .strategy = &sysctl_string,

+ .strategy = &sysctl_uts_string,

},

{

.ctl_name = KERN_OSRELEASE,

@@ -266,7 +270,7 @@ static ctl_table kern_table[] = {

.maxlen = sizeof(init_uts_ns.name.release),

.mode = 0444,

.proc_handler = &proc_do_uts_string,

- .strategy = &sysctl_string,

+ .strategy = &sysctl_uts_string,

},

{

```

        .ctl_name = KERN_VERSION,
@@ -275,7 +279,7 @@ static ctl_table kern_table[] = {
    .maxlen = sizeof(init_uts_ns.name.version),
    .mode = 0444,
    .proc_handler = &proc_do_uts_string,
- .strategy = &sysctl_string,
+ .strategy = &sysctl_uts_string,
    },
    {
        .ctl_name = KERN_NODENAME,
@@ -284,7 +288,7 @@ static ctl_table kern_table[] = {
    .maxlen = sizeof(init_uts_ns.name.nodename),
    .mode = 0644,
    .proc_handler = &proc_do_uts_string,
- .strategy = &sysctl_string,
+ .strategy = &sysctl_uts_string,
    },
    {
        .ctl_name = KERN_DOMAINNAME,
@@ -293,7 +297,7 @@ static ctl_table kern_table[] = {
    .maxlen = sizeof(init_uts_ns.name.domainname),
    .mode = 0644,
    .proc_handler = &proc_do_uts_string,
- .strategy = &sysctl_string,
+ .strategy = &sysctl_uts_string,
    },
    {
        .ctl_name = KERN_PANIC,
@@ -2600,6 +2604,23 @@ int sysctl_ms_jiffies(ctl_table *table,
    return 1;
}

+
+/* The generic string strategy routine: */
+static int sysctl_uts_string(ctl_table *table, int __user *name, int nlen,
+    void __user *oldval, size_t __user *oldlenp,
+    void __user *newval, size_t newlen, void **context)
+{
+    struct ctl_table uts_table;
+    int r, write;
+    write = newval && newlen;
+    memcpy(&uts_table, table, sizeof(uts_table));
+    uts_table.data = get_uts(table, write);
+    r = sysctl_string(&uts_table, name, nlen,
+        oldval, oldlenp, newval, newlen, context);
+    put_uts(table, write, uts_table.data);
+    return r;
+}

```

```

+
#else /* CONFIG_SYSCTL_SYSCALL */

@@ -2664,6 +2685,12 @@ int sysctl_ms_jiffies(ctl_table *table,
    return -ENOSYS;
}

+static int sysctl_uts_string(ctl_table *table, int __user *name, int nlen,
+ void __user *oldval, size_t __user *oldlenp,
+ void __user *newval, size_t newlen, void **context)
+{
+ return -ENOSYS;
+}
#endif /* CONFIG_SYSCTL_SYSCALL */

/*
--
1.4.2.rc3.g7e18e-dirty

```

Containers mailing list
Containers@lists.osdl.org
<https://lists.osdl.org/mailman/listinfo/containers>
