Subject: [PATCH] Remove CTL_UNNUMBERED
Posted by Alexey Dobriyan on Thu, 26 Jul 2007 16:45:18 GMT
View Forum Message <> Reply to Message

CTL_UNNUMBERED is unneeded, because it expands to

.ctl_name = 0

The same effect can be achieved by skipping .ctl_name initialization,
saving one line per sysctl.

Update docs and headers telling people to not add CTL_ numbers and
giving example.

This is probably all we can do to stop the flow of new CTL_ numbers,
because most of sysctls are copy-pasted. CTL_UNNUMBERED doesn't solve
this problem at all.

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

```
 Documentation/sysctl/ctl_unnumbered.txt    |   15 +++++++++++-----
 arch/ia64/kernel/crash.c                   |    1 -
 arch/ia64/kernel/perfmon.c                 |    5 -----
 arch/ia64/sn/kernel/xpc_main.c             |    5 -----
 arch/mips/au1000/common/power.c            |    4 ----
 arch/sh64/kernel/traps.c                   |    5 -----
 drivers/char/hpet.c                        |    2 --
 drivers/char/rtc.c                         |    2 --
 fs/coda/sysctl.c                           |    4 ----
 fs/lockd/svc.c                             |    7 -------
 fs/nfs/sysctl.c                            |    5 -----
 fs/ntfs/sysctl.c                           |    1 -
 include/linux/sysctl.h                     |    6 ++----
 kernel/sysctl.c                            |   20 --------------------
 net/9p/sysctl.c                            |    2 --
 net/core/sysctl_net_core.c                 |    2 --
 net/ipv6/addrconf.c                        |    1 -
 net/netfilter/nf_conntrack_proto_udplite.c |    2 --
 net/netfilter/nf_conntrack_standalone.c    |    1 -
 19 files changed, 12 insertions(+), 78 deletions(-)
```

--- a/Documentation/sysctl/ctl_unnumbered.txt
+++ b/Documentation/sysctl/ctl_unnumbered.txt
@@ -3,10 +3,6 @@ Except for a few extremely rare exceptions user space applications do not use
 the binary sysctl interface. Instead everyone uses /proc/sys/... with
 readable ascii names.

```
-Recently the kernel has started supporting setting the binary sysctl value to
-CTL_UNNUMBERED so we no longer need to assign a binary sysctl path to allow
-sysctls to show up in /proc/sys.
-
 Assigning binary sysctl numbers is an endless source of conflicts in sysctl.h,
 breaking of the user space ABI (because of those conflicts), and maintenance
 problems.  A complete pass through all of the sysctl users revealed multiple
@@ -14,7 +10,16 @@ instances where the sysctl binary interface was broken and had gone
undetected
 for years.

 So please do not add new binary sysctl numbers.  They are unneeded and
-problematic.
+problematic. Instead, use C99 initalizers, skip .ctl_name, and initialize only
+.procname:
+
+ {
+  .procname = "print-fatal-signals",
+  .data  = &print_fatal_signals,
+  .maxlen  = sizeof(int),
+  .mode  = 0644,
+  .proc_handler = &proc_dointvec,
+ },

 If you really need a new binary sysctl number please first merge your sysctl
 into the kernel and then as a separate patch allocate a binary sysctl number.
--- a/arch/ia64/kernel/crash.c
+++ b/arch/ia64/kernel/crash.c
@@ -197,7 +197,6 @@ kdump_init_notifier(struct notifier_block *self, unsigned long val, void
*data)
 #ifdef CONFIG_SYSCTL
 static ctl_table kdump_on_init_table[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "kdump_on_init",
   .data = &kdump_on_init,
   .maxlen = sizeof(int),
--- a/arch/ia64/kernel/perfmon.c
+++ b/arch/ia64/kernel/perfmon.c
@@ -521,7 +521,6 @@ EXPORT_SYMBOL(pfm_sysctl);

 static ctl_table pfm_ctl_table[]={
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "debug",
   .data  = &pfm_sysctl.debug,
   .maxlen  = sizeof(int),
```

```
@@ -529,7 +528,6 @@ static ctl_table pfm_ctl_table[]={
   .proc_handler = &proc_dointvec,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "debug_ovfl",
   .data  = &pfm_sysctl.debug_ovfl,
   .maxlen  = sizeof(int),
@@ -537,7 +535,6 @@ static ctl_table pfm_ctl_table[]={
   .proc_handler = &proc_dointvec,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "fastctxsw",
   .data  = &pfm_sysctl.fastctxsw,
   .maxlen  = sizeof(int),
@@ -545,7 +542,6 @@ static ctl_table pfm_ctl_table[]={
   .proc_handler =  &proc_dointvec,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "expert_mode",
   .data  = &pfm_sysctl.expert_mode,
   .maxlen  = sizeof(int),
@@ -556,7 +552,6 @@ static ctl_table pfm_ctl_table[]={
 };
 static ctl_table pfm_sysctl_dir[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "perfmon",
   .mode  = 0755,
   .child  = pfm_ctl_table,
--- a/arch/ia64/sn/kernel/xpc_main.c
+++ b/arch/ia64/sn/kernel/xpc_main.c
@@ -101,7 +101,6 @@ static int xpc_disengage_request_max_timelimit = 120;

 static ctl_table xpc_sys_xpc_hb_dir[] = {
  {
-  .ctl_name  = CTL_UNNUMBERED,
   .procname = "hb_interval",
   .data  = &xpc_hb_interval,
   .maxlen  = sizeof(int),
@@ -112,7 +111,6 @@ static ctl_table xpc_sys_xpc_hb_dir[] = {
   .extra2  = &xpc_hb_max_interval
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "hb_check_interval",
```

```
   .data  = &xpc_hb_check_interval,
   .maxlen  = sizeof(int),
@@ -126,13 +124,11 @@ static ctl_table xpc_sys_xpc_hb_dir[] = {
 };
 static ctl_table xpc_sys_xpc_dir[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "hb",
   .mode  = 0555,
   .child  = xpc_sys_xpc_hb_dir
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "disengage_request_timelimit",
   .data  = &xpc_disengage_request_timelimit,
   .maxlen  = sizeof(int),
@@ -146,7 +142,6 @@ static ctl_table xpc_sys_xpc_dir[] = {
 };
 static ctl_table xpc_sys_dir[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "xpc",
   .mode  = 0555,
   .child  = xpc_sys_xpc_dir
--- a/arch/mips/au1000/common/power.c
+++ b/arch/mips/au1000/common/power.c
@@ -420,7 +420,6 @@ static int pm_do_freq(ctl_table * ctl, int write, struct file *file,

 static struct ctl_table pm_table[] = {
  {
-  .ctl_name  = CTL_UNNUMBERED,
   .procname = "suspend",
   .data  = NULL,
   .maxlen  = 0,
@@ -428,7 +427,6 @@ static struct ctl_table pm_table[] = {
   .proc_handler = &pm_do_suspend
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "sleep",
   .data  = NULL,
   .maxlen  = 0,
@@ -436,7 +434,6 @@ static struct ctl_table pm_table[] = {
   .proc_handler = &pm_do_sleep
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "freq",
```

```
   .data  = NULL,
   .maxlen  = 0,
@@ -448,7 +445,6 @@ static struct ctl_table pm_table[] = {

 static struct ctl_table pm_dir_table[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "pm",
   .mode  = 0555,
   .child  = pm_table
--- a/arch/sh64/kernel/traps.c
+++ b/arch/sh64/kernel/traps.c
@@ -910,7 +910,6 @@ static int misaligned_fixup(struct pt_regs *regs)

 static ctl_table unaligned_table[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "kernel_reports",
   .data  = &kernel_mode_unaligned_fixup_count,
   .maxlen  = sizeof(int),
@@ -919,7 +918,6 @@ static ctl_table unaligned_table[] = {
  },
 #if defined(CONFIG_SH64_USER_MISALIGNED_FIXUP)
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "user_reports",
   .data  = &user_mode_unaligned_fixup_count,
   .maxlen  = sizeof(int),
@@ -927,7 +925,6 @@ static ctl_table unaligned_table[] = {
   .proc_handler = &proc_dointvec
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "user_enable",
   .data  = &user_mode_unaligned_fixup_enable,
   .maxlen  = sizeof(int),
@@ -939,7 +936,6 @@ static ctl_table unaligned_table[] = {

 static ctl_table unaligned_root[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "unaligned_fixup",
   .mode  = 0555,
   unaligned_table
@@ -949,7 +945,6 @@ static ctl_table unaligned_root[] = {

 static ctl_table sh64_root[] = {
  {
```

```
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "sh64",
   .mode  = 0555,
   .child  = unaligned_root
--- a/drivers/char/hpet.c
+++ b/drivers/char/hpet.c
@@ -723,7 +723,6 @@ int hpet_control(struct hpet_task *tp, unsigned int cmd, unsigned long
arg)

 static ctl_table hpet_table[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "max-user-freq",
   .data = &hpet_max_freq,
   .maxlen = sizeof(int),
@@ -735,7 +734,6 @@ static ctl_table hpet_table[] = {

 static ctl_table hpet_root[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "hpet",
   .maxlen = 0,
   .mode = 0555,
--- a/drivers/char/rtc.c
+++ b/drivers/char/rtc.c
@@ -279,7 +279,6 @@ irqreturn_t rtc_interrupt(int irq, void *dev_id)
  */
 static ctl_table rtc_table[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "max-user-freq",
   .data  = &rtc_max_user_freq,
   .maxlen  = sizeof(int),
@@ -291,7 +290,6 @@ static ctl_table rtc_table[] = {

 static ctl_table rtc_root[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "rtc",
   .mode  = 0555,
   .child  = rtc_table,
--- a/fs/coda/sysctl.c
+++ b/fs/coda/sysctl.c
@@ -15,7 +15,6 @@ static struct ctl_table_header *fs_table_header;

 static ctl_table coda_table[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
```

```
   .procname = "timeout",
   .data  = &coda_timeout,
   .maxlen  = sizeof(int),
@@ -23,7 +22,6 @@ static ctl_table coda_table[] = {
   .proc_handler = &proc_dointvec
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "hard",
   .data  = &coda_hard,
   .maxlen  = sizeof(int),
@@ -31,7 +29,6 @@ static ctl_table coda_table[] = {
   .proc_handler = &proc_dointvec
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "fake_statfs",
   .data  = &coda_fake_statfs,
   .maxlen  = sizeof(int),
@@ -43,7 +40,6 @@ static ctl_table coda_table[] = {

 static ctl_table fs_table[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "coda",
   .mode  = 0555,
   .child  = coda_table
--- a/fs/lockd/svc.c
+++ b/fs/lockd/svc.c
@@ -370,7 +370,6 @@ EXPORT_SYMBOL(lockd_down);

 static ctl_table nlm_sysctls[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nlm_grace_period",
   .data  = &nlm_grace_period,
   .maxlen  = sizeof(unsigned long),
@@ -380,7 +379,6 @@ static ctl_table nlm_sysctls[] = {
   .extra2  = (unsigned long *) &nlm_grace_period_max,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nlm_timeout",
   .data  = &nlm_timeout,
   .maxlen  = sizeof(unsigned long),
@@ -390,7 +388,6 @@ static ctl_table nlm_sysctls[] = {
   .extra2  = (unsigned long *) &nlm_timeout_max,
  },
```

```
 {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nlm_udpport",
   .data  = &nlm_udpport,
   .maxlen  = sizeof(int),
@@ -400,7 +397,6 @@ static ctl_table nlm_sysctls[] = {
   .extra2  = (int *) &nlm_port_max,
 },
 {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nlm_tcpport",
   .data  = &nlm_tcpport,
   .maxlen  = sizeof(int),
@@ -410,7 +406,6 @@ static ctl_table nlm_sysctls[] = {
   .extra2  = (int *) &nlm_port_max,
 },
 {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nsm_use_hostnames",
   .data  = &nsm_use_hostnames,
   .maxlen  = sizeof(int),
@@ -418,7 +413,6 @@ static ctl_table nlm_sysctls[] = {
   .proc_handler = &proc_dointvec,
 },
 {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nsm_local_state",
   .data  = &nsm_local_state,
   .maxlen  = sizeof(int),
@@ -430,7 +424,6 @@ static ctl_table nlm_sysctls[] = {

 static ctl_table nlm_sysctl_dir[] = {
 {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nfs",
   .mode  = 0555,
   .child  = nlm_sysctls,
--- a/fs/nfs/sysctl.c
+++ b/fs/nfs/sysctl.c
@@ -22,7 +22,6 @@ static struct ctl_table_header *nfs_callback_sysctl_table;
 static ctl_table nfs_cb_sysctls[] = {
 #ifdef CONFIG_NFS_V4
 {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nfs_callback_tcpport",
   .data = &nfs_callback_set_tcpport,
   .maxlen = sizeof(int),
@@ -32,7 +31,6 @@ static ctl_table nfs_cb_sysctls[] = {
```

```
   .extra2 = (int *)&nfs_set_port_max,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "idmap_cache_timeout",
   .data = &nfs_idmap_cache_timeout,
   .maxlen = sizeof(int),
@@ -42,7 +40,6 @@ static ctl_table nfs_cb_sysctls[] = {
  },
 #endif
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nfs_mountpoint_timeout",
   .data  = &nfs_mountpoint_expiry_timeout,
   .maxlen  = sizeof(nfs_mountpoint_expiry_timeout),
@@ -51,7 +48,6 @@ static ctl_table nfs_cb_sysctls[] = {
   .strategy = &sysctl_jiffies,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nfs_congestion_kb",
   .data  = &nfs_congestion_kb,
   .maxlen  = sizeof(nfs_congestion_kb),
@@ -63,7 +59,6 @@ static ctl_table nfs_cb_sysctls[] = {

 static ctl_table nfs_cb_sysctl_dir[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nfs",
   .mode = 0555,
   .child = nfs_cb_sysctls,
--- a/fs/ntfs/sysctl.c
+++ b/fs/ntfs/sysctl.c
@@ -36,7 +36,6 @@
 /* Definition of the ntfs sysctl. */
 static ctl_table ntfs_sysctls[] = {
  {
-  .ctl_name = CTL_UNNUMBERED, /* Binary and text IDs. */
   .procname = "ntfs-debug",
   .data  = &debug_msgs,  /* Data pointer and size. */
   .maxlen  = sizeof(debug_msgs),
--- a/include/linux/sysctl.h
+++ b/include/linux/sysctl.h
@@ -15,8 +15,7 @@
  **  The kernel will then return -ENOTDIR to any application using
  **  the old binary interface.
  **
- **  For new interfaces unless you really need a binary number
```

```
- **  please use CTL_UNNUMBERED.
+ **  Do not initialize .ctl_name for new sysctls.
  **
   ********************************************************** ****
   ********************************************************** ****
@@ -54,7 +53,6 @@ struct __sysctl_args {
 /* For internal pattern-matching use only: */
 #ifdef __KERNEL__
 #define CTL_NONE 0
-#define CTL_UNNUMBERED CTL_NONE /* sysctl without a binary number */
 #endif

 enum
@@ -1021,7 +1019,7 @@ extern ctl_handler sysctl_ms_jiffies;
 /* A sysctl table is an array of struct ctl_table: */
 struct ctl_table
 {
- int ctl_name;   /* Binary ID */
+ int ctl_name;  /* Binary ID, do not use in new sysctls */
  const char *procname;  /* Text ID for /proc/sys, or zero */
  void *data;
  int maxlen;
--- a/kernel/sysctl.c
+++ b/kernel/sysctl.c
@@ -222,7 +222,6 @@ static unsigned long max_wakeup_granularity_ns = 1000000000; /* 1
second */
 static ctl_table kern_table[] = {
 #ifdef CONFIG_SCHED_DEBUG
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "sched_granularity_ns",
   .data  = &sysctl_sched_granularity,
   .maxlen  = sizeof(unsigned int),
@@ -233,7 +232,6 @@ static ctl_table kern_table[] = {
   .extra2  = &max_sched_granularity_ns,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "sched_wakeup_granularity_ns",
   .data  = &sysctl_sched_wakeup_granularity,
   .maxlen  = sizeof(unsigned int),
@@ -244,7 +242,6 @@ static ctl_table kern_table[] = {
   .extra2  = &max_wakeup_granularity_ns,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "sched_batch_wakeup_granularity_ns",
   .data  = &sysctl_sched_batch_wakeup_granularity,
```

```
   .maxlen  = sizeof(unsigned int),
@@ -255,7 +252,6 @@ static ctl_table kern_table[] = {
   .extra2  = &max_wakeup_granularity_ns,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "sched_stat_granularity_ns",
   .data  = &sysctl_sched_stat_granularity,
   .maxlen  = sizeof(unsigned int),
@@ -266,7 +262,6 @@ static ctl_table kern_table[] = {
   .extra2  = &max_wakeup_granularity_ns,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "sched_runtime_limit_ns",
   .data  = &sysctl_sched_runtime_limit,
   .maxlen  = sizeof(unsigned int),
@@ -277,7 +272,6 @@ static ctl_table kern_table[] = {
   .extra2  = &max_sched_granularity_ns,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "sched_child_runs_first",
   .data  = &sysctl_sched_child_runs_first,
   .maxlen  = sizeof(unsigned int),
@@ -286,7 +280,6 @@ static ctl_table kern_table[] = {
  },
 #ifdef CONFIG_PROVE_LOCKING
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "prove_locking",
   .data  = &prove_locking,
   .maxlen  = sizeof(int),
@@ -296,7 +289,6 @@ static ctl_table kern_table[] = {
 #endif
 #ifdef CONFIG_LOCK_STAT
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "lock_stat",
   .data  = &lock_stat,
   .maxlen  = sizeof(int),
@@ -305,7 +297,6 @@ static ctl_table kern_table[] = {
  },
 #endif
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "sched_features",
   .data  = &sysctl_sched_features,
```

```
   .maxlen  = sizeof(unsigned int),
@@ -331,7 +322,6 @@ static ctl_table kern_table[] = {
  },
 #ifdef CONFIG_AUDITSYSCALL
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "audit_argv_kb",
   .data  = &audit_argv_kb,
   .maxlen  = sizeof(int),
@@ -377,7 +367,6 @@ static ctl_table kern_table[] = {
  },
 #endif
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "print-fatal-signals",
   .data  = &print_fatal_signals,
   .maxlen  = sizeof(int),
@@ -661,7 +650,6 @@ static ctl_table kern_table[] = {
   .proc_handler = &proc_dointvec,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "kstack_depth_to_print",
   .data  = &kstack_depth_to_print,
   .maxlen  = sizeof(int),
@@ -731,7 +719,6 @@ static ctl_table kern_table[] = {
 #endif
 #ifdef CONFIG_PROC_FS
  {
-  .ctl_name       = CTL_UNNUMBERED,
   .procname       = "maps_protect",
   .data           = &maps_protect,
   .maxlen         = sizeof(int),
@@ -740,7 +727,6 @@ static ctl_table kern_table[] = {
  },
 #endif
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "poweroff_cmd",
   .data  = &poweroff_cmd,
   .maxlen  = POWEROFF_CMD_PATH_LEN,
@@ -872,7 +858,6 @@ static ctl_table vm_table[] = {
   .proc_handler = &proc_dointvec,
   },
   {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "hugepages_treat_as_movable",
   .data  = &hugepages_treat_as_movable,
```

```
   .maxlen  = sizeof(int),
@@ -1005,7 +990,6 @@ static ctl_table vm_table[] = {
 #endif
 #ifdef CONFIG_SMP
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "stat_interval",
   .data  = &sysctl_stat_interval,
   .maxlen  = sizeof(sysctl_stat_interval),
@@ -1016,7 +1000,6 @@ static ctl_table vm_table[] = {
 #endif
 #ifdef CONFIG_SECURITY
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "mmap_min_addr",
   .data  = &mmap_min_addr,
   .maxlen         = sizeof(unsigned long),
@@ -1025,7 +1008,6 @@ static ctl_table vm_table[] = {
  },
 #ifdef CONFIG_NUMA
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "numa_zonelist_order",
   .data  = &numa_zonelist_order,
   .maxlen  = NUMA_ZONELIST_ORDER_LEN,
@@ -1189,7 +1171,6 @@ static ctl_table fs_table[] = {
  },
 #if defined(CONFIG_BINFMT_MISC) || defined(CONFIG_BINFMT_MISC_MODULE)
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "binfmt_misc",
   .mode  = 0555,
   .child  = binfmt_misc_table,
@@ -1205,7 +1186,6 @@ static ctl_table fs_table[] = {
 static ctl_table debug_table[] = {
 #ifdef CONFIG_X86
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "exception-trace",
   .data  = &show_unhandled_signals,
   .maxlen  = sizeof(int),
--- a/net/9p/sysctl.c
+++ b/net/9p/sysctl.c
@@ -31,7 +31,6 @@
 static struct ctl_table p9_table[] = {
 #ifdef CONFIG_NET_9P_DEBUG
  {
-  .ctl_name       = CTL_UNNUMBERED,
```

```
   .procname       = "debug",
   .data           = &p9_debug_level,
   .maxlen         = sizeof(int),
@@ -44,7 +43,6 @@ static struct ctl_table p9_table[] = {

 static struct ctl_table p9_net_table[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "9p",
   .maxlen  = 0,
   .mode  = 0555,
--- a/net/core/sysctl_net_core.c
+++ b/net/core/sysctl_net_core.c
@@ -121,7 +121,6 @@ ctl_table core_table[] = {
   .proc_handler = &proc_dointvec
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "xfrm_larval_drop",
   .data  = &sysctl_xfrm_larval_drop,
   .maxlen  = sizeof(int),
@@ -129,7 +128,6 @@ ctl_table core_table[] = {
   .proc_handler = &proc_dointvec
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "xfrm_acq_expires",
   .data  = &sysctl_xfrm_acq_expires,
   .maxlen  = sizeof(int),
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -4029,7 +4029,6 @@ static struct addrconf_sysctl_table
   },
 #ifdef CONFIG_IPV6_OPTIMISTIC_DAD
   {
-   .ctl_name = CTL_UNNUMBERED,
    .procname       =       "optimistic_dad",
    .data           =       &ipv6_devconf.optimistic_dad,
    .maxlen         =       sizeof(int),
--- a/net/netfilter/nf_conntrack_proto_udplite.c
+++ b/net/netfilter/nf_conntrack_proto_udplite.c
@@ -169,7 +169,6 @@ static unsigned int udplite_sysctl_table_users;
 static struct ctl_table_header *udplite_sysctl_header;
 static struct ctl_table udplite_sysctl_table[] = {
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nf_conntrack_udplite_timeout",
   .data  = &nf_ct_udplite_timeout,
```

```
   .maxlen  = sizeof(unsigned int),
@@ -177,7 +176,6 @@ static struct ctl_table udplite_sysctl_table[] = {
   .proc_handler = &proc_dointvec_jiffies,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nf_conntrack_udplite_timeout_stream",
   .data  = &nf_ct_udplite_timeout_stream,
   .maxlen  = sizeof(unsigned int),
--- a/net/netfilter/nf_conntrack_standalone.c
+++ b/net/netfilter/nf_conntrack_standalone.c
@@ -366,7 +366,6 @@ static ctl_table nf_ct_sysctl_table[] = {
   .extra2  = &log_invalid_proto_max,
  },
  {
-  .ctl_name = CTL_UNNUMBERED,
   .procname = "nf_conntrack_expect_max",
   .data  = &nf_ct_expect_max,
   .maxlen  = sizeof(int),
```