
Subject: [PATCH RFC 7/31] net: Make /proc/net per network namespace
Posted by [ebiederm](#) on Thu, 25 Jan 2007 19:00:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

This patch makes /proc/net per network namespace. It modifies the global variables `proc_net` and `proc_net_stat` to be per network namespace. The `proc_net` file helpers are modified to take a network namespace argument, and all of their callers are fixed to pass `init_net()` for that argument. This ensures that all of the /proc/net files are only visible and usable in the initial network namespace until the code behind them has been updated to be handle multiple network namespaces.

Making /proc/net per namespace is necessary as at least some files in /proc/net depend upon the set of network devices which is per network namespace, and even more files in /proc/net have contents that are relevant to a single network namespace.

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

drivers/isdn/divert/divert_procfs.c		7 +-
drivers/isdn/hardware/eicon/diva_didd.c		5 +-
drivers/isdn/hysdn/hysdn_proconf.c		4 +-
drivers/net/bonding/bond_main.c		7 +-
drivers/net/hamradio/bpqether.c		5 +-
drivers/net/hamradio/scc.c		4 +-
drivers/net/hamradio/yam.c		5 +-
drivers/net/ibmveth.c		6 +-
drivers/net/pppoe.c		5 +-
drivers/net/tc35815.c		1 -
drivers/net/tokenring/lanstreamer.c		4 +-
drivers/net/tokenring/olympic.c		9 +-
drivers/net/wireless/hostap/hostap_main.c		7 +-
drivers/net/wireless/strip.c		5 +-
fs/proc/Makefile		1 +
fs/proc/internal.h		5 +
fs/proc/proc_net.c		126 ++++++
fs/proc/root.c		8 +-
include/linux/proc_fs.h		28 +++-
include/net/net_namespace.h		11 ++
net/802/tr.c		3 +-
net/8021q/vlanproc.c		5 +-
net/appletalk/atalk_proc.c		7 +-
net/atm/proc.c		5 +-
net/ax25/af_ax25.c		13 +-
net/core/dev.c		9 +-
net/core/dev_mcast.c		3 +-

net/core/neighbour.c		3 +-	
net/core/pktgen.c		9 +-	
net/core/sock.c		3 +-	
net/core/wireless.c		3 +-	
net/dccp/probe.c		7 +-	
net/decnet/af_decnet.c		5 +-	
net/decnet/dn_dev.c		5 +-	
net/decnet/dn_neigh.c		5 +-	
net/decnet/dn_route.c		5 +-	
net/ieee80211/ieee80211_module.c		6 +-	
net/ipv4/arp.c		3 +-	
net/ipv4/fib_hash.c		5 +-	
net/ipv4/fib_trie.c		17 ++--	
net/ipv4/igmp.c		5 +-	
net/ipv4/ipconfig.c		3 +-	
net/ipv4/ipmr.c		5 +-	
net/ipv4/ipvs/ip_vs_app.c		5 +-	
net/ipv4/ipvs/ip_vs_conn.c		5 +-	
net/ipv4/ipvs/ip_vs_ctl.c		9 +-	
net/ipv4/ipvs/ip_vs_lblcr.c		4 +-	
net/ipv4/netfilter/ip_conntrack_standalone.c		16 ++--	
net/ipv4/netfilter/ip_queue.c		7 +-	
net/ipv4/netfilter/ipt_CLUSTERIP.c		3 +-	
net/ipv4/netfilter/ipt_recent.c		5 +-	
.../netfilter/nf_conntrack_l3proto_ipv4_compat.c		17 ++--	
net/ipv4/proc.c		11 +-	
net/ipv4/raw.c		5 +-	
net/ipv4/route.c		7 +-	
net/ipv4/tcp_ipv4.c		5 +-	
net/ipv4/tcp_probe.c		6 +-	
net/ipv4/udp.c		5 +-	
net/ipv6/addrconf.c		7 +-	
net/ipv6/anycast.c		5 +-	
net/ipv6/ip6_flowlabel.c		5 +-	
net/ipv6/mcast.c		9 +-	
net/ipv6/netfilter/ip6_queue.c		7 +-	
net/ipv6/proc.c		17 ++--	
net/ipv6/raw.c		5 +-	
net/ipv6/route.c		9 +-	
net/ipx/ipx_proc.c		7 +-	
net/irda/irproc.c		5 +-	
net/key/af_key.c		5 +-	
net/lc/lc_proc.c		7 +-	
net/netfilter/core.c		3 +-	
net/netfilter/nf_conntrack_standalone.c		19 ++--	
net/netfilter/x_tables.c		17 ++--	
net/netfilter/xt_hashlimit.c		11 +-	
net/netlink/af_netlink.c		3 +-	

```

net/netrom/af_netrom.c          | 13 +-
net/packet/af_packet.c          |  5 +-
net/rose/af_rose.c              | 17 ++--
net/rxrpc/proc.c                |  7 +-
net/sched/sch_api.c             |  3 +-
net/sctp/protocol.c             |  5 +-
net/sunrpc/stats.c              |  5 +-
net/unix/af_unix.c              |  5 +-
net/wanrouter/wanproc.c         |  7 +-
net/x25/x25_proc.c              |  7 +-
85 files changed, 462 insertions(+), 250 deletions(-)

```

```
diff --git a/drivers/isdn/divert/divert_procfs.c b/drivers/isdn/divert/divert_procfs.c
```

```
index 06967da..6517dd5 100644
```

```
--- a/drivers/isdn/divert/divert_procfs.c
```

```
+++ b/drivers/isdn/divert/divert_procfs.c
```

```
@@ -18,6 +18,7 @@
```

```
#include <linux/fs.h>
```

```
#endif
```

```
#include <linux/isdnif.h>
```

```
+#include <net/net_namespace.h>
```

```
#include "isdn_divert.h"
```

```
@@ -285,12 +286,12 @@ divert_dev_init(void)
```

```
    init_waitqueue_head(&rd_queue);
```

```
#ifdef CONFIG_PROC_FS
```

```
- isdn_proc_entry = proc_mkdir("net/isdn", NULL);
```

```
+ isdn_proc_entry = proc_mkdir("isdn", per_net(proc_net, init_net()));
```

```
    if (!isdn_proc_entry)
```

```
        return (-1);
```

```
    isdn_divert_entry = create_proc_entry("divert", S_IFREG | S_IRUGO, isdn_proc_entry);
```

```
    if (!isdn_divert_entry) {
```

```
- remove_proc_entry("net/isdn", NULL);
```

```
+ remove_proc_entry("isdn", per_net(proc_net, init_net()));
```

```
    return (-1);
```

```
}
```

```
    isdn_divert_entry->proc_fops = &isdn_fops;
```

```
@@ -310,7 +311,7 @@ divert_dev_deinit(void)
```

```
#ifdef CONFIG_PROC_FS
```

```
    remove_proc_entry("divert", isdn_proc_entry);
```

```
- remove_proc_entry("net/isdn", NULL);
```

```
+ remove_proc_entry("isdn", per_net(proc_net, init_net()));
```

```
#endif /* CONFIG_PROC_FS */
```

```
    return (0);
```

```

diff --git a/drivers/isdn/hardware/eicon/diva_didd.c b/drivers/isdn/hardware/eicon/diva_didd.c
index 14298b8..1b7c0f9 100644
--- a/drivers/isdn/hardware/eicon/diva_didd.c
+++ b/drivers/isdn/hardware/eicon/diva_didd.c
@@ -15,6 +15,7 @@
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/proc_fs.h>
+#include <net/net_namespace.h>

#include "platform.h"
#include "di_defs.h"
@@ -86,7 +87,7 @@ proc_read(char *page, char **start, off_t off, int count, int *eof,

static int DIVA_INIT_FUNCTION create_proc(void)
{
- proc_net_eicon = proc_mkdir("net/eicon", NULL);
+ proc_net_eicon = proc_mkdir("eicon", per_net(proc_net, init_net()));

if (proc_net_eicon) {
if ((proc_didd =
@@ -102,7 +103,7 @@ static int DIVA_INIT_FUNCTION create_proc(void)
static void DIVA_EXIT_FUNCTION remove_proc(void)
{
remove_proc_entry(DRIVERLNAME, proc_net_eicon);
- remove_proc_entry("net/eicon", NULL);
+ remove_proc_entry("eicon", per_net(proc_net, init_net()));
}

static int DIVA_INIT_FUNCTION divadidd_init(void)
diff --git a/drivers/isdn/hysdn/hysdn_proconf.c b/drivers/isdn/hysdn/hysdn_proconf.c
index 94a9350..b634e67 100644
--- a/drivers/isdn/hysdn/hysdn_proconf.c
+++ b/drivers/isdn/hysdn/hysdn_proconf.c
@@ -392,7 +392,7 @@ hysdn_proconf_init(void)
hysdn_card *card;
unsigned char conf_name[20];

- hysdn_proc_entry = proc_mkdir(PROC_SUBDIR_NAME, proc_net);
+ hysdn_proc_entry = proc_mkdir(PROC_SUBDIR_NAME, per_net(proc_net, init_net()));
if (!hysdn_proc_entry) {
printk(KERN_ERR "HYSDN: unable to create hysdn subdir\n");
return (-1);
@@ -437,5 +437,5 @@ hysdn_proconf_release(void)
card = card->next; /* point to next card */
}

- remove_proc_entry(PROC_SUBDIR_NAME, proc_net);

```

```

+ remove_proc_entry(PROC_SUBDIR_NAME, per_net(proc_net, init_net()));
}
diff --git a/drivers/net/bonding/bond_main.c b/drivers/net/bonding/bond_main.c
index 6482aed..9b3bf4e 100644
--- a/drivers/net/bonding/bond_main.c
+++ b/drivers/net/bonding/bond_main.c
@@ -75,6 +75,7 @@
#include <linux/if_vlan.h>
#include <linux/if_bonding.h>
#include <net/route.h>
+#include <net/net_namespace.h>
#include "bonding.h"
#include "bond_3ad.h"
#include "bond_alb.h"
@@ -3169,7 +3170,7 @@ static void bond_create_proc_dir(void)
{
    int len = strlen(DRV_NAME);

- for (bond_proc_dir = proc_net->subdir; bond_proc_dir;
+ for (bond_proc_dir = per_net(proc_net, init_net())->subdir; bond_proc_dir;
      bond_proc_dir = bond_proc_dir->next) {
    if ((bond_proc_dir->namelen == len) &&
        !memcmp(bond_proc_dir->name, DRV_NAME, len)) {
@@ -3178,7 +3179,7 @@ static void bond_create_proc_dir(void)
}

if (!bond_proc_dir) {
- bond_proc_dir = proc_mkdir(DRV_NAME, proc_net);
+ bond_proc_dir = proc_mkdir(DRV_NAME, per_net(proc_net, init_net()));
    if (bond_proc_dir) {
        bond_proc_dir->owner = THIS_MODULE;
    } else {
@@ -3213,7 +3214,7 @@ static void bond_destroy_proc_dir(void)
    bond_proc_dir->owner = NULL;
}
} else {
- remove_proc_entry(DRV_NAME, proc_net);
+ remove_proc_entry(DRV_NAME, per_net(proc_net, init_net()));
    bond_proc_dir = NULL;
}
}
diff --git a/drivers/net/hamradio/bpqether.c b/drivers/net/hamradio/bpqether.c
index 5b788d8..9fc92ad 100644
--- a/drivers/net/hamradio/bpqether.c
+++ b/drivers/net/hamradio/bpqether.c
@@ -83,6 +83,7 @@
#include <net/ip.h>

```

```

#include <net/arp.h>
#include <net/net_namespace.h>

#include <linux/bpqether.h>

@@ -594,7 +595,7 @@ static int bpq_device_event(struct notifier_block *this,unsigned long
event, voi
static int __init bpq_init_driver(void)
{
#ifdef CONFIG_PROC_FS
- if (!proc_net_fops_create("bpqether", S_IRUGO, &bpq_info_fops)) {
+ if (!proc_net_fops_create(init_net(), "bpqether", S_IRUGO, &bpq_info_fops)) {
    printk(KERN_ERR
        "bpq: cannot create /proc/net/bpqether entry.\n");
    return -ENOENT;
@@ -618,7 +619,7 @@ static void __exit bpq_cleanup_driver(void)

    unregister_netdevice_notifier(&bpq_dev_notifier);

- proc_net_remove("bpqether");
+ proc_net_remove(init_net(), "bpqether");

    rtnl_lock();
    while (!list_empty(&bpq_devices)) {
diff --git a/drivers/net/hamradio/scc.c b/drivers/net/hamradio/scc.c
index 2ce047e..2000597 100644
--- a/drivers/net/hamradio/scc.c
+++ b/drivers/net/hamradio/scc.c
@@ -2114,7 +2114,7 @@ static int __init scc_init_driver (void)
}
rtnl_unlock();

- proc_net_fops_create("z8530drv", 0, &scc_net_seq_fops);
+ proc_net_fops_create(init_net(), "z8530drv", 0, &scc_net_seq_fops);

    return 0;
}
@@ -2169,7 +2169,7 @@ static void __exit scc_cleanup_driver(void)
if (Vector_Latch)
    release_region(Vector_Latch, 1);

- proc_net_remove("z8530drv");
+ proc_net_remove(init_net(), "z8530drv");
}

MODULE_AUTHOR("Joerg Reuter <jreuter@yaina.de>");
diff --git a/drivers/net/hamradio/yam.c b/drivers/net/hamradio/yam.c
index 6d74f08..3e92f3b 100644

```

```

--- a/drivers/net/hamradio/yam.c
+++ b/drivers/net/hamradio/yam.c
@@ -60,6 +60,7 @@
#include <linux/etherdevice.h>
#include <linux/skbuff.h>
#include <net/ax25.h>
+#include <net/net_namespace.h>

#include <linux/kernel.h>
#include <linux/proc_fs.h>
@@ -1147,7 +1148,7 @@ static int __init yam_init_driver(void)
    yam_timer.expires = jiffies + HZ / 100;
    add_timer(&yam_timer);

- proc_net_fops_create("yam", S_IRUGO, &yam_info_fops);
+ proc_net_fops_create(init_net(), "yam", S_IRUGO, &yam_info_fops);
    return 0;
    error:
    while (--i >= 0) {
@@ -1179,7 +1180,7 @@ static void __exit yam_cleanup_driver(void)
    kfree(p);
}

- proc_net_remove("yam");
+ proc_net_remove(init_net(), "yam");
}

/* ----- */
diff --git a/drivers/net/ibmveth.c b/drivers/net/ibmveth.c
index 99343b5..d8b0ba8 100644
--- a/drivers/net/ibmveth.c
+++ b/drivers/net/ibmveth.c
@@ -97,7 +97,7 @@ static inline void ibmveth_rxq_harvest_buffer(struct ibmveth_adapter
*adapter);
static struct kobj_type ktype_veth_pool;

#ifdef CONFIG_PROC_FS
-#define IBMVETH_PROC_DIR "net/ibmveth"
+#define IBMVETH_PROC_DIR "ibmveth"
static struct proc_dir_entry *ibmveth_proc_dir;
#endif

@@ -1073,7 +1073,7 @@ static int __devexit ibmveth_remove(struct vio_dev *dev)
#ifdef CONFIG_PROC_FS
static void ibmveth_proc_register_driver(void)
{
- ibmveth_proc_dir = proc_mkdir(IBMVETH_PROC_DIR, NULL);
+ ibmveth_proc_dir = proc_mkdir(IBMVETH_PROC_DIR, per_net(proc_net, init_net()));

```

```

    if (ibmveth_proc_dir) {
        SET_MODULE_OWNER(ibmveth_proc_dir);
    }
@@ -1081,7 +1081,7 @@ static void ibmveth_proc_register_driver(void)

static void ibmveth_proc_unregister_driver(void)
{
- remove_proc_entry(IBMETH_PROC_DIR, NULL);
+ remove_proc_entry(IBMETH_PROC_DIR, per_net(proc_net, init_net()));
}

static void *ibmveth_seq_start(struct seq_file *seq, loff_t *pos)
diff --git a/drivers/net/pppoe.c b/drivers/net/pppoe.c
index 315d5c3..d34fe16 100644
--- a/drivers/net/pppoe.c
+++ b/drivers/net/pppoe.c
@@ -72,6 +72,7 @@
#include <linux/proc_fs.h>
#include <linux/seq_file.h>

+#include <net/net_namespace.h>
#include <net/sock.h>

#include <asm/uaccess.h>
@@ -1055,7 +1056,7 @@ static int __init pppoe_proc_init(void)
{
    struct proc_dir_entry *p;

- p = create_proc_entry("net/pppoe", S_IRUGO, NULL);
+ p = create_proc_entry("pppoe", S_IRUGO, per_net(proc_net, init_net()));
    if (!p)
        return -ENOMEM;

@@ -1126,7 +1127,7 @@ static void __exit pppoe_exit(void)
    dev_remove_pack(&pppoe_ptype);
    dev_remove_pack(&pppoe_d_ptype);
    unregister_netdevice_notifier(&pppoe_notifier);
- remove_proc_entry("net/pppoe", NULL);
+ remove_proc_entry("pppoe", per_net(proc_net, init_net()));
    proto_unregister(&pppoe_sk_proto);
}

diff --git a/drivers/net/tc35815.c b/drivers/net/tc35815.c
index 81ed82f..1f26c29 100644
--- a/drivers/net/tc35815.c
+++ b/drivers/net/tc35815.c
@@ -61,7 +61,6 @@ static const char *version =
    * io regions, irqs and dma channels

```

```

*/
static const char* cardname = "TC35815CF";
#define TC35815_PROC_ENTRY "net/tc35815"

#define TC35815_MODULE_NAME "TC35815CF"
#define TX_TIMEOUT (4*HZ)
diff --git a/drivers/net/tokenring/lanstreamer.c b/drivers/net/tokenring/lanstreamer.c
index e999feb..b382ef3 100644
--- a/drivers/net/tokenring/lanstreamer.c
+++ b/drivers/net/tokenring/lanstreamer.c
@@ -250,7 +250,7 @@ static int __devinit streamer_init_one(struct pci_dev *pdev,
#ifdef STREAMER_NETWORK_MONITOR
#ifdef CONFIG_PROC_FS
    if (!dev_streamer)
-   create_proc_read_entry("net/streamer_tr", 0, 0,
+   create_proc_read_entry("streamer_tr", 0, per_net(proc_net, init_net()),
        streamer_proc_info, NULL);
    streamer_priv->next = dev_streamer;
    dev_streamer = streamer_priv;
@@ -423,7 +423,7 @@ static void __devexit streamer_remove_one(struct pci_dev *pdev)
    }
    if (!dev_streamer)
-   remove_proc_entry("net/streamer_tr", NULL);
+   remove_proc_entry("streamer_tr", per_net(proc_net, init_net()));
    }
#endif
#endif
diff --git a/drivers/net/tokenring/olympic.c b/drivers/net/tokenring/olympic.c
index 8f4ecc1..6b74c3b 100644
--- a/drivers/net/tokenring/olympic.c
+++ b/drivers/net/tokenring/olympic.c
@@ -101,6 +101,7 @@
#include <linux/bitops.h>
#include <linux/jiffies.h>

+#include <net/net_namespace.h>
#include <net/checksum.h>

#include <asm/io.h>
@@ -268,9 +269,9 @@ static int __devinit olympic_probe(struct pci_dev *pdev, const struct
pci_device
    printk("Olympic: %s registered as: %s\n", olympic_priv->olympic_card_name, dev->name);
    if (olympic_priv->olympic_network_monitor) { /* Must go after register_netdev as we need the
device name */
        char proc_name[20];
-   strcpy(proc_name, "net/olympic_");
+   strcpy(proc_name, "olympic_");

```

```

    strcat(proc_name,dev->name) ;
- create_proc_read_entry(proc_name,0,NULL,olympic_proc_info,(void *)dev) ;
+ create_proc_read_entry(proc_name,0,per_net(proc_net, init_net()),olympic_proc_info,(void
*)dev) ;
    printk("Olympic: Network Monitor information: /proc/%s\n",proc_name);
}
return 0 ;
@@ -1750,9 +1751,9 @@ static void __devexit olympic_remove_one(struct pci_dev *pdev)

    if (olympic_priv->olympic_network_monitor) {
        char proc_name[20] ;
- strcpy(proc_name,"net/olympic_") ;
+ strcpy(proc_name,"olympic_") ;
        strcat(proc_name,dev->name) ;
- remove_proc_entry(proc_name,NULL);
+ remove_proc_entry(proc_name,per_net(proc_net, init_net()));
    }
    unregister_netdev(dev) ;
    iounmap(olympic_priv->olympic_mmio) ;
diff --git a/drivers/net/wireless/hostap/hostap_main.c b/drivers/net/wireless/hostap/hostap_main.c
index 04c19ce..69b56d6 100644
--- a/drivers/net/wireless/hostap/hostap_main.c
+++ b/drivers/net/wireless/hostap/hostap_main.c
@@ -24,6 +24,7 @@
#include <linux/rtnetlink.h>
#include <linux/wireless.h>
#include <linux/etherdevice.h>
+#include <net/net_namespace.h>
#include <net/iw_handler.h>
#include <net/ieee80211.h>
#include <net/ieee80211_crypt.h>
@@ -1093,8 +1094,8 @@ struct proc_dir_entry *hostap_proc;

static int __init hostap_init(void)
{
- if (proc_net != NULL) {
- hostap_proc = proc_mkdir("hostap", proc_net);
+ if (per_net(proc_net, init_net()) != NULL) {
+ hostap_proc = proc_mkdir("hostap", per_net(proc_net, init_net()));
    if (!hostap_proc)
        printk(KERN_WARNING "Failed to mkdir "
            "/proc/net/hostap\n");
@@ -1109,7 +1110,7 @@ static void __exit hostap_exit(void)
{
    if (hostap_proc != NULL) {
        hostap_proc = NULL;
- remove_proc_entry("hostap", proc_net);
+ remove_proc_entry("hostap", per_net(proc_net, init_net()));

```

```
}  
}
```

```
diff --git a/drivers/net/wireless/strip.c b/drivers/net/wireless/strip.c  
index ce3a8ba..6c27ff2 100644  
--- a/drivers/net/wireless/strip.c  
+++ b/drivers/net/wireless/strip.c  
@@ -107,6 +107,7 @@ static const char StripVersion[] = "1.3A-STUART.CHESHIRE";  
#include <linux/serialP.h>  
#include <linux/rcupdate.h>  
#include <net/arp.h>  
+#include <net/net_namespace.h>  
  
#include <linux/ip.h>  
#include <linux/tcp.h>  
@@ -2789,7 +2790,7 @@ static int __init strip_init_driver(void)  
/*  
 * Register the status file with /proc  
 */  
- proc_net_fops_create("strip", S_IFREG | S_IRUGO, &strip_seq_fops);  
+ proc_net_fops_create(init_net(), "strip", S_IFREG | S_IRUGO, &strip_seq_fops);  
  
return status;  
}  
@@ -2811,7 +2812,7 @@ static void __exit strip_exit_driver(void)  
}  
  
/* Unregister with the /proc/net file here. */  
- proc_net_remove("strip");  
+ proc_net_remove(init_net(), "strip");  
  
if ((i = tty_unregister_ldisc(N_STRIP)))  
    printk(KERN_ERR "STRIP: can't unregister line discipline (err = %d)\n", i);  
diff --git a/fs/proc/Makefile b/fs/proc/Makefile  
index a6b3a8f..63cc3ce 100644  
--- a/fs/proc/Makefile  
+++ b/fs/proc/Makefile  
@@ -10,6 +10,7 @@ proc-$(CONFIG_MMU) := mmu.o task_mmu.o  
proc-y      += inode.o root.o base.o generic.o array.o \  
    proc_tty.o proc_misc.o proc_sysctl.o  
  
+proc-$(CONFIG_NET) += proc_net.o  
proc-$(CONFIG_PROC_KCORE) += kcore.o  
proc-$(CONFIG_PROC_VMCORE) += vmcore.o  
proc-$(CONFIG_PROC_DEVICETREE) += proc_devtree.o  
diff --git a/fs/proc/internal.h b/fs/proc/internal.h  
index 3c9a305..f916252 100644  
--- a/fs/proc/internal.h
```

```

+++ b/fs/proc/internal.h
@@ -12,6 +12,11 @@
#include <linux/proc_fs.h>

extern int proc_sys_init(void);
#ifdef CONFIG_NET
extern int proc_net_init(void);
#else
+static inline int proc_net_init(void) { return 0; }
#endif

struct vmalloc_info {
    unsigned long used;
diff --git a/fs/proc/proc_net.c b/fs/proc/proc_net.c
new file mode 100644
index 0000000..022dd9a
--- /dev/null
+++ b/fs/proc/proc_net.c
@@ -0,0 +1,126 @@
+/*
+ * linux/fs/proc/net.c
+ *
+ * Copyright (C) 2007
+ *
+ * Author: Eric Biederman <ebiederm@xmission.com>
+ *
+ * proc net directory handling functions
+ */
+
+#include <asm/uaccess.h>
+
+#include <linux/errno.h>
+#include <linux/time.h>
+#include <linux/proc_fs.h>
+#include <linux/stat.h>
+#include <linux/init.h>
+#include <linux/sched.h>
+#include <linux/module.h>
+#include <linux/bitops.h>
+#include <linux/smp_lock.h>
+#include <linux/mount.h>
+#include <linux/nsproxy.h>
+#include <net/net_namespace.h>
+
+#include "internal.h"
+
+static struct proc_dir_entry *proc_net_shadow;
+DEFINE_PER_NET(struct proc_dir_entry *, proc_net);

```

```

+DEFINE_PER_NET(struct proc_dir_entry *, proc_net_stat);
+EXPORT_PER_NET_SYMBOL(proc_net);
+EXPORT_PER_NET_SYMBOL(proc_net_stat);
+
+static DEFINE_PER_NET(struct proc_dir_entry, proc_net_root);
+
+static struct dentry *proc_net_shadow_dentry(struct dentry *parent,
+      struct proc_dir_entry *de)
+{
+ struct dentry *shadow = NULL;
+ struct inode *inode;
+ if (!de)
+   goto out;
+ inode = proc_get_inode(parent->d_inode->i_sb, de->low_ino, de);
+ if (!inode)
+   goto out;
+ shadow = d_alloc_name(parent, de->name);
+ if (!shadow)
+   goto out_iput;
+ shadow->d_op = parent->d_op; /* proc_dentry_operations */
+ d_instantiate(shadow, inode);
+out:
+ return shadow;
+out_iput:
+ iput(inode);
+ goto out;
+}
+
+static void *proc_net_follow_link(struct dentry *parent, struct nameidata *nd)
+{
+ net_t net = current->nsproxy->net_ns;
+ struct dentry *shadow;
+ shadow = proc_net_shadow_dentry(parent, per_net(proc_net, net));
+ if (!shadow)
+   return ERR_PTR(-ENOENT);
+
+ dput(nd->dentry);
+ /* My dentry count is 1 and that should be enough as the
+  * shadow dentry is thrown away immediately.
+  */
+ nd->dentry = shadow;
+ return NULL;
+}
+
+static const struct file_operations proc_net_dir_operations = {
+ .read = generic_read_dir,
+};
+

```

```

+static struct inode_operations proc_net_dir_inode_operations = {
+ .follow_link = proc_net_follow_link,
+};
+
+
+static int proc_net_ns_init(net_t net)
+{
+ struct proc_dir_entry *netd, *net_statd;
+
+ netd = proc_mkdir("net", &per_net(proc_net_root, net));
+ if (!netd)
+ return -EEXIST;
+
+ net_statd = proc_mkdir("stat", netd);
+ if (!net_statd) {
+ remove_proc_entry("net", &per_net(proc_net_root, net));
+ return -EEXIST;
+ }
+
+ netd->data = net_to_voidp(net);
+ net_statd->data = net_to_voidp(net);
+ per_net(proc_net_root, net).data = net_to_voidp(net);
+
+ per_net(proc_net, net) = netd;
+ per_net(proc_net_stat, net) = net_statd;
+
+ return 0;
+}
+
+static void proc_net_ns_exit(net_t net)
+{
+ remove_proc_entry("stat", per_net(proc_net, net));
+ remove_proc_entry("net", &per_net(proc_net_root, net));
+
+}
+
+struct pernet_operations proc_net_ns_ops = {
+ .init = proc_net_ns_init,
+ .exit = proc_net_ns_exit,
+};
+
+int proc_net_init(void)
+{
+ proc_net_shadow = proc_mkdir("net", NULL);
+ proc_net_shadow->proc_iops = &proc_net_dir_inode_operations;
+ proc_net_shadow->proc_fops = &proc_net_dir_operations;
+
+ return register_pernet_subsys(&proc_net_ns_ops);

```

```

+}
diff --git a/fs/proc/root.c b/fs/proc/root.c
index 4d42406..7c3939c 100644
--- a/fs/proc/root.c
+++ b/fs/proc/root.c
@@ -21,7 +21,7 @@

#include "internal.h"

-struct proc_dir_entry *proc_net, *proc_net_stat, *proc_bus, *proc_root_fs, *proc_root_driver;
+struct proc_dir_entry *proc_bus, *proc_root_fs, *proc_root_driver;

static int proc_get_sb(struct file_system_type *fs_type,
    int flags, const char *dev_name, void *data, struct vfsmount *mnt)
@@ -61,8 +61,8 @@ void __init proc_root_init(void)
    return;
}
proc_misc_init();
- proc_net = proc_mkdir("net", NULL);
- proc_net_stat = proc_mkdir("net/stat", NULL);
+
+ proc_net_init();

#ifdef CONFIG_SYSVIPC
    proc_mkdir("sysvipc", NULL);
@@ -161,7 +161,5 @@ EXPORT_SYMBOL(create_proc_entry);
EXPORT_SYMBOL(remove_proc_entry);
EXPORT_SYMBOL(proc_root);
EXPORT_SYMBOL(proc_root_fs);
-EXPORT_SYMBOL(proc_net);
-EXPORT_SYMBOL(proc_net_stat);
EXPORT_SYMBOL(proc_bus);
EXPORT_SYMBOL(proc_root_driver);
diff --git a/include/linux/proc_fs.h b/include/linux/proc_fs.h
index 2969913..c1b958d 100644
--- a/include/linux/proc_fs.h
+++ b/include/linux/proc_fs.h
@@ -5,6 +5,7 @@
@@ -5,6 +5,7 @@
#include <linux/fs.h>
#include <linux/spinlock.h>
#include <linux/magic.h>
+#include <linux/net_namespace_type.h>
#include <asm/atomic.h>

/*
@@ -85,8 +86,8 @@ struct vmcore {

extern struct proc_dir_entry proc_root;

```

```

extern struct proc_dir_entry *proc_root_fs;
-extern struct proc_dir_entry *proc_net;
-extern struct proc_dir_entry *proc_net_stat;
+DECLARE_PER_NET(struct proc_dir_entry *, proc_net);
+DECLARE_PER_NET(struct proc_dir_entry *, proc_net_stat);
extern struct proc_dir_entry *proc_bus;
extern struct proc_dir_entry *proc_root_driver;
extern struct proc_dir_entry *proc_root_kcore;
@@ -183,24 +184,25 @@ static inline struct proc_dir_entry *create_proc_info_entry(const char
*name,
    return res;
}

-static inline struct proc_dir_entry *proc_net_create(const char *name,
- mode_t mode, get_info_t *get_info)
+static inline struct proc_dir_entry *proc_net_create(net_t net,
+ const char *name, mode_t mode, get_info_t *get_info)
{
- return create_proc_info_entry(name,mode,proc_net,get_info);
+ return create_proc_info_entry(name,mode,per_net(proc_net, net),get_info);
}

-static inline struct proc_dir_entry *proc_net_fops_create(const char *name,
- mode_t mode, const struct file_operations *fops)
+static inline struct proc_dir_entry *proc_net_fops_create(net_t net,
+ const char *name, mode_t mode, const struct file_operations *fops)
{
- struct proc_dir_entry *res = create_proc_entry(name, mode, proc_net);
+ struct proc_dir_entry *res =
+ create_proc_entry(name, mode, per_net(proc_net, net));
    if (res)
        res->proc_fops = fops;
    return res;
}

-static inline void proc_net_remove(const char *name)
+static inline void proc_net_remove(net_t net, const char *name)
{
- remove_proc_entry(name,proc_net);
+ remove_proc_entry(name, per_net(proc_net, net));
}

#else
@@ -209,9 +211,9 @@ static inline void proc_net_remove(const char *name)
#define proc_net NULL
#define proc_bus NULL

-#define proc_net_fops_create(name, mode, fops) ({ (void)(mode), NULL; })

```

```

-#define proc_net_create(name, mode, info) ({ (void)(mode), NULL; })
-static inline void proc_net_remove(const char *name) {}
+#define proc_net_fops_create(net, name, mode, fops) ({ (void)(mode), NULL; })
+#define proc_net_create(net, name, mode, info) ({ (void)(mode), NULL; })
+static inline void proc_net_remove(net_t net, const char *name) {}

static inline void proc_flush_task(struct task_struct *task) { }

diff --git a/include/net/net_namespace.h b/include/net/net_namespace.h
index 9208e2e..b64568f 100644
--- a/include/net/net_namespace.h
+++ b/include/net/net_namespace.h
@@ -8,6 +8,7 @@
#include <linux/workqueue.h>
#include <linux/nsproxy.h>
#include <linux/net_namespace_type.h>
+#include <linux/proc_fs.h>

/* How many bytes in each network namespace should we allocate
 * for use by modules when they are loaded.
@@ -65,4 +66,14 @@ extern void unregister_pernet_subsys(struct pernet_operations *);
extern int register_pernet_device(struct pernet_operations *);
extern void unregister_pernet_device(struct pernet_operations *);

+static inline net_t PDE_NET(struct proc_dir_entry *pde)
+{
+ return net_from_voidp(pde->parent->data);
+}
+
+static inline net_t PROC_NET(const struct inode *inode)
+{
+ return PDE_NET(PDE(inode));
+}
+
#endif /* __NET_NET_NAMESPACE_H */
diff --git a/net/802/tr.c b/net/802/tr.c
index 829deb4..3324fa6 100644
--- a/net/802/tr.c
+++ b/net/802/tr.c
@@ -36,6 +36,7 @@
#include <linux/seq_file.h>
#include <linux/init.h>
#include <net/arp.h>
+#include <net/net_namespace.h>

static void tr_add_rif_info(struct trh_hdr *trh, struct net_device *dev);
static void rif_check_expire(unsigned long dummy);
@@ -636,7 +637,7 @@ static int __init rif_init(void)

```

```

rif_timer.function = rif_check_expire;
add_timer(&rif_timer);

- proc_net_fops_create("tr_rif", S_IRUGO, &rif_seq_fops);
+ proc_net_fops_create(init_net(), "tr_rif", S_IRUGO, &rif_seq_fops);
  return 0;
}

diff --git a/net/8021q/vlanproc.c b/net/8021q/vlanproc.c
index a8fc0de..abcf58c 100644
--- a/net/8021q/vlanproc.c
+++ b/net/8021q/vlanproc.c
@@ -33,6 +33,7 @@
#include <linux/fs.h>
#include <linux/netdevice.h>
#include <linux/if_vlan.h>
+#include <net/net_namespace.h>
#include "vlanproc.h"
#include "vlan.h"

@@ -143,7 +144,7 @@ void vlan_proc_cleanup(void)
  remove_proc_entry(name_conf, proc_vlan_dir);

  if (proc_vlan_dir)
-  proc_net_remove(name_root);
+  proc_net_remove(init_net(), name_root);

  /* Dynamically added entries should be cleaned up as their vlan_device
   * is removed, so we should not have to take care of it here...
@@ -156,7 +157,7 @@ void vlan_proc_cleanup(void)

int __init vlan_proc_init(void)
{
-  proc_vlan_dir = proc_mkdir(name_root, proc_net);
+  proc_vlan_dir = proc_mkdir(name_root, per_net(proc_net, init_net()));
  if (proc_vlan_dir) {
    proc_vlan_conf = create_proc_entry(name_conf,
      S_IFREG|S_IRUSR|S_IWUSR,
diff --git a/net/appletalk/atalk_proc.c b/net/appletalk/atalk_proc.c
index 7ae4916..0e77c68 100644
--- a/net/appletalk/atalk_proc.c
+++ b/net/appletalk/atalk_proc.c
@@ -13,6 +13,7 @@
#include <linux/seq_file.h>
#include <net/sock.h>
#include <linux/atalk.h>
+#include <net/net_namespace.h>

```

```

static __inline__ struct atalk_iface *atalk_get_interface_idx(loff_t pos)
@@ -271,7 +272,7 @@ int __init atalk_proc_init(void)
    struct proc_dir_entry *p;
    int rc = -ENOMEM;

- atalk_proc_dir = proc_mkdir("atalk", proc_net);
+ atalk_proc_dir = proc_mkdir("atalk", per_net(proc_net, init_net()));
    if (!atalk_proc_dir)
        goto out;
    atalk_proc_dir->owner = THIS_MODULE;
@@ -306,7 +307,7 @@ out_socket:
out_route:
    remove_proc_entry("interface", atalk_proc_dir);
out_interface:
- remove_proc_entry("atalk", proc_net);
+ remove_proc_entry("atalk", per_net(proc_net, init_net()));
    goto out;
}

@@ -316,5 +317,5 @@ void __exit atalk_proc_exit(void)
    remove_proc_entry("route", atalk_proc_dir);
    remove_proc_entry("socket", atalk_proc_dir);
    remove_proc_entry("arp", atalk_proc_dir);
- remove_proc_entry("atalk", proc_net);
+ remove_proc_entry("atalk", per_net(proc_net, init_net()));
}
diff --git a/net/atm/proc.c b/net/atm/proc.c
index 739866b..8b0299d 100644
--- a/net/atm/proc.c
+++ b/net/atm/proc.c
@@ -22,6 +22,7 @@
#include <linux/netdevice.h>
#include <linux/atmclip.h>
#include <linux/init.h> /* for __init */
+#include <net/net_namespace.h>
#include <net/atmclip.h>
#include <asm/uaccess.h>
#include <asm/atomic.h>
@@ -475,7 +476,7 @@ static void atm_proc_dirs_remove(void)
    if (e->dirent)
        remove_proc_entry(e->name, atm_proc_root);
}
- remove_proc_entry("net/atm", NULL);
+ remove_proc_entry("atm", per_net(proc_net, init_net()));
}

int __init atm_proc_init(void)

```

```

@@ -483,7 +484,7 @@ int __init atm_proc_init(void)
    static struct atm_proc_entry *e;
    int ret;

- atm_proc_root = proc_mkdir("net/atm",NULL);
+ atm_proc_root = proc_mkdir("atm", per_net(proc_net, init_net()));
    if (!atm_proc_root)
        goto err_out;
    for (e = atm_proc_ents; e->name; e++) {
diff --git a/net/ax25/af_ax25.c b/net/ax25/af_ax25.c
index 42233df..e60af4e 100644
--- a/net/ax25/af_ax25.c
+++ b/net/ax25/af_ax25.c
@@ -48,6 +48,7 @@
#include <net/tcp_states.h>
#include <net/ip.h>
#include <net/arp.h>
+#include <net/net_namespace.h>

@@ -2000,9 +2001,9 @@ static int __init ax25_init(void)
    register_netdevice_notifier(&ax25_dev_notifier);
    ax25_register_sysctl();

- proc_net_fops_create("ax25_route", S_IRUGO, &ax25_route_fops);
- proc_net_fops_create("ax25", S_IRUGO, &ax25_info_fops);
- proc_net_fops_create("ax25_calls", S_IRUGO, &ax25_uid_fops);
+ proc_net_fops_create(init_net(), "ax25_route", S_IRUGO, &ax25_route_fops);
+ proc_net_fops_create(init_net(), "ax25", S_IRUGO, &ax25_info_fops);
+ proc_net_fops_create(init_net(), "ax25_calls", S_IRUGO, &ax25_uid_fops);
    out:
    return rc;
}
@@ -2016,9 +2017,9 @@ MODULE_ALIAS_NETPROTO(PF_AX25);

static void __exit ax25_exit(void)
{
- proc_net_remove("ax25_route");
- proc_net_remove("ax25");
- proc_net_remove("ax25_calls");
+ proc_net_remove(init_net(), "ax25_route");
+ proc_net_remove(init_net(), "ax25");
+ proc_net_remove(init_net(), "ax25_calls");
    ax25_rt_free();
    ax25_uid_free();
    ax25_dev_free();
diff --git a/net/core/dev.c b/net/core/dev.c

```

index 17c07f3..90e4c0e 100644

--- a/net/core/dev.c

+++ b/net/core/dev.c

@@ -116,6 +116,7 @@

#include <linux/dmaengine.h>

#include <linux/err.h>

#include <linux/ctype.h>

+#include <net/net_namespace.h>

/*

* The list of packet types we will receive (as opposed to discard)

@@ -2238,9 +2239,9 @@ static int __init dev_proc_init(void)

{

int rc = -ENOMEM;

- if (!proc_net_fops_create("dev", S_IRUGO, &dev_seq_fops))

+ if (!proc_net_fops_create(init_net(), "dev", S_IRUGO, &dev_seq_fops))
goto out;

- if (!proc_net_fops_create("softnet_stat", S_IRUGO, &softnet_seq_fops))

+ if (!proc_net_fops_create(init_net(), "softnet_stat", S_IRUGO, &softnet_seq_fops))
goto out_dev;

if (wireless_proc_init())

goto out_softnet;

@@ -2248,9 +2249,9 @@ static int __init dev_proc_init(void)

out:

return rc;

out_softnet:

- proc_net_remove("softnet_stat");

+ proc_net_remove(init_net(), "softnet_stat");

out_dev:

- proc_net_remove("dev");

+ proc_net_remove(init_net(), "dev");

goto out;

}

#else

diff --git a/net/core/dev_mcast.c b/net/core/dev_mcast.c

index b22648d..623e606 100644

--- a/net/core/dev_mcast.c

+++ b/net/core/dev_mcast.c

@@ -47,6 +47,7 @@

#include <linux/skbuff.h>

#include <net/sock.h>

#include <net/arp.h>

+#include <net/net_namespace.h>

/*

@@ -289,7 +290,7 @@ static struct file_operations dev_mc_seq_fops = {

```

void __init dev_mcast_init(void)
{
- proc_net_fops_create("dev_mcast", 0, &dev_mc_seq_fops);
+ proc_net_fops_create(init_net(), "dev_mcast", 0, &dev_mc_seq_fops);
}

EXPORT_SYMBOL(dev_mc_add);
diff --git a/net/core/neighbour.c b/net/core/neighbour.c
index 8437678..90e1d2e 100644
--- a/net/core/neighbour.c
+++ b/net/core/neighbour.c
@@ -34,6 +34,7 @@
#include <linux/rtnetlink.h>
#include <linux/random.h>
#include <linux/string.h>
+#include <net/net_namespace.h>

#define NEIGH_DEBUG 1

@@ -1348,7 +1349,7 @@ void neigh_table_init_no_netlink(struct neigh_table *tbl)
    panic("cannot create neighbour cache statistics");

#ifdef CONFIG_PROC_FS
- tbl->pde = create_proc_entry(tbl->id, 0, proc_net_stat);
+ tbl->pde = create_proc_entry(tbl->id, 0, per_net(proc_net_stat, init_net()));
    if (!tbl->pde)
        panic("cannot create neighbour proc dir entry");
    tbl->pde->proc_fops = &neigh_stat_seq_fops;
diff --git a/net/core/pktgen.c b/net/core/pktgen.c
index 04d4b93..ab48533 100644
--- a/net/core/pktgen.c
+++ b/net/core/pktgen.c
@@ -152,6 +152,7 @@
#include <net/checksum.h>
#include <net/ipv6.h>
#include <net/addrconf.h>
+#include <net/net_namespace.h>
#include <asm/byteorder.h>
#include <linux/rcupdate.h>
#include <asm/bitops.h>
@@ -3565,7 +3566,7 @@ static int __init pg_init(void)

    printk(version);

- pg_proc_dir = proc_mkdir(PG_PROC_DIR, proc_net);
+ pg_proc_dir = proc_mkdir(PG_PROC_DIR, per_net(proc_net, init_net()));
    if (!pg_proc_dir)

```

```

    return -ENODEV;
    pg_proc_dir->owner = THIS_MODULE;
@@ -3574,7 +3575,7 @@ static int __init pg_init(void)
    if (pe == NULL) {
        printk("pktgen: ERROR: cannot create %s procfs entry.\n",
            PGCTRL);
-   proc_net_remove(PG_PROC_DIR);
+   proc_net_remove(init_net(), PG_PROC_DIR);
        return -EINVAL;
    }

@@ -3597,7 +3598,7 @@ static int __init pg_init(void)
    printk("pktgen: ERROR: Initialization failed for all threads\n");
    unregister_netdevice_notifier(&pktgen_notifier_block);
    remove_proc_entry(PGCTRL, pg_proc_dir);
-   proc_net_remove(PG_PROC_DIR);
+   proc_net_remove(init_net(), PG_PROC_DIR);
    return -ENODEV;
}

@@ -3624,7 +3625,7 @@ static void __exit pg_cleanup(void)

    /* Clean up proc file system */
    remove_proc_entry(PGCTRL, pg_proc_dir);
-   proc_net_remove(PG_PROC_DIR);
+   proc_net_remove(init_net(), PG_PROC_DIR);
}

module_init(pg_init);
diff --git a/net/core/sock.c b/net/core/sock.c
index 0ed5b4f..5555364 100644
--- a/net/core/sock.c
+++ b/net/core/sock.c
@@ -123,6 +123,7 @@
#include <net/sock.h>
#include <net/xfrm.h>
#include <linux/ipsec.h>
+#include <net/net_namespace.h>

#include <linux/filter.h>

@@ -1922,7 +1923,7 @@ static struct file_operations proto_seq_fops = {
static int __init proto_init(void)
{
    /* register /proc/net/protocols */
-   return proc_net_fops_create("protocols", S_IRUGO, &proto_seq_fops) == NULL ? -ENOBUFFS :
0;
+   return proc_net_fops_create(init_net(), "protocols", S_IRUGO, &proto_seq_fops) == NULL ?

```

```

-ENOBUFS : 0;
}

subsys_initcall(proto_init);
diff --git a/net/core/wireless.c b/net/core/wireless.c
index f69ab7b..faa242f 100644
--- a/net/core/wireless.c
+++ b/net/core/wireless.c
@@ -94,6 +94,7 @@
#include <linux/wireless.h> /* Pretty obvious */
#include <net/iw_handler.h> /* New driver API */
#include <net/netlink.h>
+#include <net/net_namespace.h>

#include <asm/uaccess.h> /* copy_to_user() */

@@ -685,7 +686,7 @@ static struct file_operations wireless_seq_fops = {
int __init wireless_proc_init(void)
{
/* Create /proc/net/wireless entry */
- if (!proc_net_fops_create("wireless", S_IRUGO, &wireless_seq_fops))
+ if (!proc_net_fops_create(init_net(), "wireless", S_IRUGO, &wireless_seq_fops))
return -ENOMEM;

return 0;
diff --git a/net/dccp/probe.c b/net/dccp/probe.c
index f81e37d..7c1c1ef 100644
--- a/net/dccp/probe.c
+++ b/net/dccp/probe.c
@@ -30,6 +30,7 @@
#include <linux/module.h>
#include <linux/kfifo.h>
#include <linux/vmalloc.h>
+#include <net/net_namespace.h>

#include "dccp.h"
#include "ccid.h"
@@ -165,7 +166,7 @@ static __init int dccpprobe_init(void)
if (IS_ERR(dccpw.fifo))
return PTR_ERR(dccpw.fifo);

- if (!proc_net_fops_create(procname, S_IRUSR, &dccpprobe_fops))
+ if (!proc_net_fops_create(init_net(), procname, S_IRUSR, &dccpprobe_fops))
goto err0;

ret = register_jprobe(&dccp_send_probe);
@@ -175,7 +176,7 @@ static __init int dccpprobe_init(void)
pr_info("DCCP watch registered (port=%d)\n", port);

```

```

    return 0;
err1:
- proc_net_remove(procname);
+ proc_net_remove(init_net(), procname);
err0:
    kfifo_free(dccpw.fifo);
    return ret;
@@ -185,7 +186,7 @@ module_init(dccpprobe_init);
static __exit void dccpprobe_exit(void)
{
    kfifo_free(dccpw.fifo);
- proc_net_remove(procname);
+ proc_net_remove(init_net(), procname);
    unregister_jprobe(&dccp_send_probe);

}
diff --git a/net/decnet/af_decnet.c b/net/decnet/af_decnet.c
index 21f20f2..77cd802 100644
--- a/net/decnet/af_decnet.c
+++ b/net/decnet/af_decnet.c
@@ -131,6 +131,7 @@ Version 0.0.6 2.1.110 07-aug-98 Eduardo Marcelo Serrat
#include <net/neighbour.h>
#include <net/dst.h>
#include <net/fib_rules.h>
+#include <net/net_namespace.h>
#include <net/dn.h>
#include <net/dn_nsp.h>
#include <net/dn_dev.h>
@@ -2396,7 +2397,7 @@ static int __init decnet_init(void)
    dev_add_pack(&dn_dix_packet_type);
    register_netdevice_notifier(&dn_dev_notifier);

- proc_net_fops_create("decnet", S_IRUGO, &dn_socket_seq_fops);
+ proc_net_fops_create(init_net(), "decnet", S_IRUGO, &dn_socket_seq_fops);
    dn_register_sysctl();
out:
    return rc;
@@ -2424,7 +2425,7 @@ static void __exit decnet_exit(void)
    dn_neigh_cleanup();
    dn_fib_cleanup();

- proc_net_remove("decnet");
+ proc_net_remove(init_net(), "decnet");

    proto_unregister(&dn_proto);
}
diff --git a/net/decnet/dn_dev.c b/net/decnet/dn_dev.c
index 913e25a..19b1469 100644

```

```

--- a/net/decnet/dn_dev.c
+++ b/net/decnet/dn_dev.c
@@ -47,6 +47,7 @@
#include <net/flow.h>
#include <net/fib_rules.h>
#include <net/netlink.h>
+#include <net/net_namespace.h>
#include <net/dn.h>
#include <net/dn_dev.h>
#include <net/dn_route.h>
@@ -1483,7 +1484,7 @@ void __init dn_dev_init(void)

    rtnetlink_links[PF_DECnet] = dnet_rtnetlink_table;

- proc_net_fops_create("decnet_dev", S_IRUGO, &dn_dev_seq_fops);
+ proc_net_fops_create(init_net(), "decnet_dev", S_IRUGO, &dn_dev_seq_fops);

#ifdef CONFIG_SYSCTL
{
@@ -1506,7 +1507,7 @@ void __exit dn_dev_cleanup(void)
}
#endif /* CONFIG_SYSCTL */

- proc_net_remove("decnet_dev");
+ proc_net_remove(init_net(), "decnet_dev");

    dn_dev_devices_off();
}
diff --git a/net/decnet/dn_neigh.c b/net/decnet/dn_neigh.c
index 7322bb3..fd99aca 100644
--- a/net/decnet/dn_neigh.c
+++ b/net/decnet/dn_neigh.c
@@ -38,6 +38,7 @@
#include <linux/rcupdate.h>
#include <linux/jhash.h>
#include <asm/atomic.h>
+#include <net/net_namespace.h>
#include <net/neighbour.h>
#include <net/dst.h>
#include <net/flow.h>
@@ -611,11 +612,11 @@ static struct file_operations dn_neigh_seq_fops = {
void __init dn_neigh_init(void)
{
    neigh_table_init(&dn_neigh_table);
- proc_net_fops_create("decnet_neigh", S_IRUGO, &dn_neigh_seq_fops);
+ proc_net_fops_create(init_net(), "decnet_neigh", S_IRUGO, &dn_neigh_seq_fops);
}

```

```

void __exit dn_neigh_cleanup(void)
{
- proc_net_remove("decnet_neigh");
+ proc_net_remove(init_net(), "decnet_neigh");
  neigh_table_clear(&dn_neigh_table);
}
diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index 9881933..0d657eb 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -81,6 +81,7 @@
#include <net/dst.h>
#include <net/flow.h>
#include <net/fib_rules.h>
+#include <net/net_namespace.h>
#include <net/dn.h>
#include <net/dn_dev.h>
#include <net/dn_nsp.h>
@@ -1811,7 +1812,7 @@ void __init dn_route_init(void)

    dn_dst_ops.gc_thresh = (dn_rt_hash_mask + 1);

- proc_net_fops_create("decnet_cache", S_IRUGO, &dn_rt_cache_seq_fops);
+ proc_net_fops_create(init_net(), "decnet_cache", S_IRUGO, &dn_rt_cache_seq_fops);
}

void __exit dn_route_cleanup(void)
@@ -1819,6 +1820,6 @@ void __exit dn_route_cleanup(void)
    del_timer(&dn_route_timer);
    dn_run_flush(0);

- proc_net_remove("decnet_cache");
+ proc_net_remove(init_net(), "decnet_cache");
}

diff --git a/net/ieee80211/ieee80211_module.c b/net/ieee80211/ieee80211_module.c
index b1c6d1f..23539f6 100644
--- a/net/ieee80211/ieee80211_module.c
+++ b/net/ieee80211/ieee80211_module.c
@@ -263,7 +263,7 @@ static int __init ieee80211_init(void)
    struct proc_dir_entry *e;

    ieee80211_debug_level = debug;
- ieee80211_proc = proc_mkdir(DRV_NAME, proc_net);
+ ieee80211_proc = proc_mkdir(DRV_NAME, per_net(proc_net, init_net()));
    if (ieee80211_proc == NULL) {
        IEEE80211_ERROR("Unable to create " DRV_NAME
            " proc directory\n");

```

```

@@ -272,7 +272,7 @@ static int __init ieee80211_init(void)
    e = create_proc_entry("debug_level", S_IFREG | S_IRUGO | S_IWUSR,
        ieee80211_proc);
    if (!e) {
- remove_proc_entry(DRV_NAME, proc_net);
+ remove_proc_entry(DRV_NAME, per_net(proc_net, init_net()));
        ieee80211_proc = NULL;
        return -EIO;
    }
@@ -292,7 +292,7 @@ static void __exit ieee80211_exit(void)
#ifdef CONFIG_IEEE80211_DEBUG
    if (ieee80211_proc) {
        remove_proc_entry("debug_level", ieee80211_proc);
- remove_proc_entry(DRV_NAME, proc_net);
+ remove_proc_entry(DRV_NAME, per_net(proc_net, init_net()));
        ieee80211_proc = NULL;
    }
#endif /* CONFIG_IEEE80211_DEBUG */
diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c
index 3981e8b..e3b89a7 100644
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -110,6 +110,7 @@
#include <net/protocol.h>
#include <net/tcp.h>
#include <net/sock.h>
+#include <net/net_namespace.h>
#include <net/arp.h>
#if defined(CONFIG_AX25) || defined(CONFIG_AX25_MODULE)
#include <net/ax25.h>
@@ -1400,7 +1401,7 @@ static struct file_operations arp_seq_fops = {

static int __init arp_proc_init(void)
{
- if (!proc_net_fops_create("arp", S_IRUGO, &arp_seq_fops))
+ if (!proc_net_fops_create(init_net(), "arp", S_IRUGO, &arp_seq_fops))
    return -ENOMEM;
    return 0;
}
diff --git a/net/ipv4/fib_hash.c b/net/ipv4/fib_hash.c
index 648f47c..42ea992 100644
--- a/net/ipv4/fib_hash.c
+++ b/net/ipv4/fib_hash.c
@@ -41,6 +41,7 @@
#include <net/route.h>
#include <net/tcp.h>
#include <net/sock.h>
+#include <net/net_namespace.h>

```

```

#include <net/ip_fib.h>

#include "fib_lookup.h"
@@ -1067,13 +1068,13 @@ static struct file_operations fib_seq_fops = {

int __init fib_proc_init(void)
{
- if (!proc_net_fops_create("route", S_IRUGO, &fib_seq_fops))
+ if (!proc_net_fops_create(init_net(), "route", S_IRUGO, &fib_seq_fops))
    return -ENOMEM;
    return 0;
}

void __init fib_proc_exit(void)
{
- proc_net_remove("route");
+ proc_net_remove(init_net(), "route");
}
#endif /* CONFIG_PROC_FS */
diff --git a/net/ipv4/fib_trie.c b/net/ipv4/fib_trie.c
index 13307c0..94598b3 100644
--- a/net/ipv4/fib_trie.c
+++ b/net/ipv4/fib_trie.c
@@ -79,6 +79,7 @@
#include <net/route.h>
#include <net/tcp.h>
#include <net/sock.h>
+#include <net/net_namespace.h>
#include <net/ip_fib.h>
#include "fib_lookup.h"

@@ -2494,30 +2495,30 @@ static struct file_operations fib_route_fops = {

int __init fib_proc_init(void)
{
- if (!proc_net_fops_create("fib_trie", S_IRUGO, &fib_trie_fops))
+ if (!proc_net_fops_create(init_net(), "fib_trie", S_IRUGO, &fib_trie_fops))
    goto out1;

- if (!proc_net_fops_create("fib_triestat", S_IRUGO, &fib_triestat_fops))
+ if (!proc_net_fops_create(init_net(), "fib_triestat", S_IRUGO, &fib_triestat_fops))
    goto out2;

- if (!proc_net_fops_create("route", S_IRUGO, &fib_route_fops))
+ if (!proc_net_fops_create(init_net(), "route", S_IRUGO, &fib_route_fops))
    goto out3;

    return 0;

```

```

out3:
- proc_net_remove("fib_triestat");
+ proc_net_remove(init_net(), "fib_triestat");
out2:
- proc_net_remove("fib_trie");
+ proc_net_remove(init_net(), "fib_trie");
out1:
    return -ENOMEM;
}

void __init fib_proc_exit(void)
{
- proc_net_remove("fib_trie");
- proc_net_remove("fib_triestat");
- proc_net_remove("route");
+ proc_net_remove(init_net(), "fib_trie");
+ proc_net_remove(init_net(), "fib_triestat");
+ proc_net_remove(init_net(), "route");
}

#endif /* CONFIG_PROC_FS */
diff --git a/net/ipv4/igmp.c b/net/ipv4/igmp.c
index 0017ccb..92624cc 100644
--- a/net/ipv4/igmp.c
+++ b/net/ipv4/igmp.c
@@ -97,6 +97,7 @@
#include <net/route.h>
#include <net/sock.h>
#include <net/checksum.h>
+#include <net/net_namespace.h>
#include <linux/netfilter_ipv4.h>
#ifdef CONFIG_IP_MROUTE
#include <linux/mroute.h>
@@ -2585,8 +2586,8 @@ static struct file_operations igmp_mcf_seq_fops = {

int __init igmp_mc_proc_init(void)
{
- proc_net_fops_create("igmp", S_IRUGO, &igmp_mc_seq_fops);
- proc_net_fops_create("mcfilter", S_IRUGO, &igmp_mcf_seq_fops);
+ proc_net_fops_create(init_net(), "igmp", S_IRUGO, &igmp_mc_seq_fops);
+ proc_net_fops_create(init_net(), "mcfilter", S_IRUGO, &igmp_mcf_seq_fops);
    return 0;
}
#endif
diff --git a/net/ipv4/ipconfig.c b/net/ipv4/ipconfig.c
index afa60b9..8b649c5 100644
--- a/net/ipv4/ipconfig.c

```

```

+++ b/net/ipv4/ipconfig.c
@@ -59,6 +59,7 @@
#include <net/ip.h>
#include <net/ipconfig.h>
#include <net/route.h>
+#include <net/net_namespace.h>

#include <asm/uaccess.h>
#include <net/checksum.h>
@@ -1252,7 +1253,7 @@ static int __init ip_auto_config(void)
__be32 addr;

#ifdef CONFIG_PROC_FS
- proc_net_fops_create("pnp", S_IRUGO, &pnp_seq_fops);
+ proc_net_fops_create(init_net(), "pnp", S_IRUGO, &pnp_seq_fops);
#endif /* CONFIG_PROC_FS */

if (!lic_enable)
diff --git a/net/ipv4/ipmr.c b/net/ipv4/ipmr.c
index ecb5422..af50394 100644
--- a/net/ipv4/ipmr.c
+++ b/net/ipv4/ipmr.c
@@ -63,6 +63,7 @@
#include <linux/netfilter_ipv4.h>
#include <net/ipv4.h>
#include <net/checksum.h>
+#include <net/net_namespace.h>

#if defined(CONFIG_IP_PIMSM_V1) || defined(CONFIG_IP_PIMSM_V2)
#define CONFIG_IP_PIMSM 1
@@ -1906,7 +1907,7 @@ void __init ip_mr_init(void)
ipmr_expire_timer.function=ipmr_expire_process;
register_netdevice_notifier(&ip_mr_notifier);
#ifdef CONFIG_PROC_FS
- proc_net_fops_create("ip_mr_vif", 0, &ipmr_vif_fops);
- proc_net_fops_create("ip_mr_cache", 0, &ipmr_mfc_fops);
+ proc_net_fops_create(init_net(), "ip_mr_vif", 0, &ipmr_vif_fops);
+ proc_net_fops_create(init_net(), "ip_mr_cache", 0, &ipmr_mfc_fops);
#endif
}
diff --git a/net/ipv4/ipvs/ip_vs_app.c b/net/ipv4/ipvs/ip_vs_app.c
index 6c40899..4f44452 100644
--- a/net/ipv4/ipvs/ip_vs_app.c
+++ b/net/ipv4/ipvs/ip_vs_app.c
@@ -32,6 +32,7 @@
#include <linux/proc_fs.h>
#include <linux/seq_file.h>
#include <linux/mutex.h>

```

```

+#include <net/net_namespace.h>

#include <net/ip_vs.h>

@@ -618,12 +619,12 @@ int ip_vs_skb_replace(struct sk_buff *skb, gfp_t pri,
int ip_vs_app_init(void)
{
    /* we will replace it with proc_net_ipvs_create() soon */
- proc_net_fops_create("ip_vs_app", 0, &ip_vs_app_fops);
+ proc_net_fops_create(init_net(), "ip_vs_app", 0, &ip_vs_app_fops);
    return 0;
}

void ip_vs_app_cleanup(void)
{
- proc_net_remove("ip_vs_app");
+ proc_net_remove(init_net(), "ip_vs_app");
}
diff --git a/net/ipv4/ipvs/ip_vs_conn.c b/net/ipv4/ipvs/ip_vs_conn.c
index 8086787..0764e0f 100644
--- a/net/ipv4/ipvs/ip_vs_conn.c
+++ b/net/ipv4/ipvs/ip_vs_conn.c
@@ -34,6 +34,7 @@
#include <linux/seq_file.h>
#include <linux/jhash.h>
#include <linux/random.h>
+#include <net/net_namespace.h>

#include <net/ip_vs.h>

@@ -923,7 +924,7 @@ int ip_vs_conn_init(void)
    rwlock_init(&__ip_vs_conntbl_lock_array[idx].l);
}

- proc_net_fops_create("ip_vs_conn", 0, &ip_vs_conn_fops);
+ proc_net_fops_create(init_net(), "ip_vs_conn", 0, &ip_vs_conn_fops);

/* calculate the random value for connection hash */
get_random_bytes(&ip_vs_conn_rnd, sizeof(ip_vs_conn_rnd));
@@ -939,6 +940,6 @@ void ip_vs_conn_cleanup(void)

/* Release the empty cache */
kmem_cache_destroy(ip_vs_conn_cache);
- proc_net_remove("ip_vs_conn");
+ proc_net_remove(init_net(), "ip_vs_conn");
vfree(ip_vs_conn_tab);
}

```

```

diff --git a/net/ipv4/ipvs/ip_vs_ctl.c b/net/ipv4/ipvs/ip_vs_ctl.c
index c4e4237..d4bf160 100644
--- a/net/ipv4/ipvs/ip_vs_ctl.c
+++ b/net/ipv4/ipvs/ip_vs_ctl.c
@@ -39,6 +39,7 @@
#include <net/ip.h>
#include <net/route.h>
#include <net/sock.h>
+#include <net/net_namespace.h>

#include <asm/uaccess.h>

@@ -2356,8 +2357,8 @@ int ip_vs_control_init(void)
    return ret;
}

- proc_net_fops_create("ip_vs", 0, &ip_vs_info_fops);
- proc_net_fops_create("ip_vs_stats",0, &ip_vs_stats_fops);
+ proc_net_fops_create(init_net(), "ip_vs", 0, &ip_vs_info_fops);
+ proc_net_fops_create(init_net(), "ip_vs_stats",0, &ip_vs_stats_fops);

    sysctl_header = register_sysctl_table(vs_root_table);

@@ -2389,8 +2390,8 @@ void ip_vs_control_cleanup(void)
    cancel_rearming_delayed_work(&defense_work);
    ip_vs_kill_estimator(&ip_vs_stats);
    unregister_sysctl_table(sysctl_header);
- proc_net_remove("ip_vs_stats");
- proc_net_remove("ip_vs");
+ proc_net_remove(init_net(), "ip_vs_stats");
+ proc_net_remove(init_net(), "ip_vs");
    nf_unregister_sockopt(&ip_vs_sockopts);
    LeaveFunction(2);
}
diff --git a/net/ipv4/ipvs/ip_vs_lblcr.c b/net/ipv4/ipvs/ip_vs_lblcr.c
index 22004f8..f8491e7 100644
--- a/net/ipv4/ipvs/ip_vs_lblcr.c
+++ b/net/ipv4/ipvs/ip_vs_lblcr.c
@@ -843,7 +843,7 @@ static int __init ip_vs_lblcr_init(void)
    INIT_LIST_HEAD(&ip_vs_lblcr_scheduler.n_list);
    sysctl_header = register_sysctl_table(lblcr_root_table);
#ifdef CONFIG_IP_VS_LBLCR_DEBUG
- proc_net_create("ip_vs_lblcr", 0, ip_vs_lblcr_getinfo);
+ proc_net_create(init_net(), "ip_vs_lblcr", 0, ip_vs_lblcr_getinfo);
#endif
    return register_ip_vs_scheduler(&ip_vs_lblcr_scheduler);
}
@@ -852,7 +852,7 @@ static int __init ip_vs_lblcr_init(void)

```

```

static void __exit ip_vs_lblcr_cleanup(void)
{
#ifdef CONFIG_IP_VS_LBLCR_DEBUG
- proc_net_remove("ip_vs_lblcr");
+ proc_net_remove(init_net(), "ip_vs_lblcr");
#endif
    unregister_sysctl_table(sysctl_header);
    unregister_ip_vs_scheduler(&ip_vs_lblcr_scheduler);
diff --git a/net/ipv4/netfilter/ip_conntrack_standalone.c
b/net/ipv4/netfilter/ip_conntrack_standalone.c
index 9d89469..d04cbb0 100644
--- a/net/ipv4/netfilter/ip_conntrack_standalone.c
+++ b/net/ipv4/netfilter/ip_conntrack_standalone.c
@@ -828,14 +828,14 @@ static int __init ip_conntrack_standalone_init(void)

#ifdef CONFIG_PROC_FS
    ret = -ENOMEM;
- proc = proc_net_fops_create("ip_conntrack", 0440, &ct_file_ops);
+ proc = proc_net_fops_create(init_net(), "ip_conntrack", 0440, &ct_file_ops);
    if (!proc) goto cleanup_init;

- proc_exp = proc_net_fops_create("ip_conntrack_expect", 0440,
+ proc_exp = proc_net_fops_create(init_net(), "ip_conntrack_expect", 0440,
    &exp_file_ops);
    if (!proc_exp) goto cleanup_proc;

- proc_stat = create_proc_entry("ip_conntrack", S_IRUGO, proc_net_stat);
+ proc_stat = create_proc_entry("ip_conntrack", S_IRUGO, per_net(proc_net_stat, init_net()));
    if (!proc_stat)
        goto cleanup_proc_exp;

@@ -864,11 +864,11 @@ static int __init ip_conntrack_standalone_init(void)
#endif
    cleanup_proc_stat;
#ifdef CONFIG_PROC_FS
- remove_proc_entry("ip_conntrack", proc_net_stat);
+ remove_proc_entry("ip_conntrack", per_net(proc_net_stat, init_net()));
    cleanup_proc_exp:
- proc_net_remove("ip_conntrack_expect");
+ proc_net_remove(init_net(), "ip_conntrack_expect");
    cleanup_proc:
- proc_net_remove("ip_conntrack");
+ proc_net_remove(init_net(), "ip_conntrack");
    cleanup_init:
#endif /* CONFIG_PROC_FS */
    ip_conntrack_cleanup();
@@ -884,8 +884,8 @@ static void __exit ip_conntrack_standalone_fini(void)
    nf_unregister_hooks(ip_conntrack_ops, ARRAY_SIZE(ip_conntrack_ops));

```

```

#ifdef CONFIG_PROC_FS
    remove_proc_entry("ip_conntrack", proc_net_stat);
-   proc_net_remove("ip_conntrack_expect");
-   proc_net_remove("ip_conntrack");
+   proc_net_remove(init_net(), "ip_conntrack_expect");
+   proc_net_remove(init_net(), "ip_conntrack");
#endif /* CONFIG_PROC_FS */
    ip_conntrack_cleanup();
}
diff --git a/net/ipv4/netfilter/ip_queue.c b/net/ipv4/netfilter/ip_queue.c
index 3446d4a..aae660c 100644
--- a/net/ipv4/netfilter/ip_queue.c
+++ b/net/ipv4/netfilter/ip_queue.c
@@ -38,6 +38,7 @@
#include <linux/mutex.h>
#include <net/sock.h>
#include <net/route.h>
+#include <net/net_namespace.h>

#define IPQ_QMAX_DEFAULT 1024
#define IPQ_PROC_FS_NAME "ip_queue"
@@ -684,7 +685,7 @@ static int __init ip_queue_init(void)
    goto cleanup_netlink_notifier;
}

-   proc = proc_net_create(IPQ_PROC_FS_NAME, 0, ipq_get_info);
+   proc = proc_net_create(init_net(), IPQ_PROC_FS_NAME, 0, ipq_get_info);
    if (proc)
        proc->owner = THIS_MODULE;
    else {
@@ -705,7 +706,7 @@ static int __init ip_queue_init(void)
cleanup_sysctl:
    unregister_sysctl_table(ipq_sysctl_header);
    unregister_netdevice_notifier(&ipq_dev_notifier);
-   proc_net_remove(IPQ_PROC_FS_NAME);
+   proc_net_remove(init_net(), IPQ_PROC_FS_NAME);

cleanup_ipqnl:
    sock_release(ipqnl->sk_socket);
@@ -725,7 +726,7 @@ static void __exit ip_queue_fini(void)

    unregister_sysctl_table(ipq_sysctl_header);
    unregister_netdevice_notifier(&ipq_dev_notifier);
-   proc_net_remove(IPQ_PROC_FS_NAME);
+   proc_net_remove(init_net(), IPQ_PROC_FS_NAME);

    sock_release(ipqnl->sk_socket);
    mutex_lock(&ipqnl_mutex);

```

```

diff --git a/net/ipv4/netfilter/ipt_CLUSTERIP.c b/net/ipv4/netfilter/ipt_CLUSTERIP.c
index b1c1116..779e2c6 100644
--- a/net/ipv4/netfilter/ipt_CLUSTERIP.c
+++ b/net/ipv4/netfilter/ipt_CLUSTERIP.c
@@ -22,6 +22,7 @@
#include <linux/proc_fs.h>
#include <linux/seq_file.h>

+#include <net/net_namespace.h>
#include <net/checksum.h>

#include <linux/netfilter_arp.h>
@@ -736,7 +737,7 @@ static int __init ipt_clusterip_init(void)
    goto cleanup_target;

#ifdef CONFIG_PROC_FS
- clusterip_procdir = proc_mkdir("ipt_CLUSTERIP", proc_net);
+ clusterip_procdir = proc_mkdir("ipt_CLUSTERIP", per_net(proc_net, init_net()));
    if (!clusterip_procdir) {
        printk(KERN_ERR "CLUSTERIP: Unable to proc dir entry\n");
        ret = -ENOMEM;
diff --git a/net/ipv4/netfilter/ipt_recent.c b/net/ipv4/netfilter/ipt_recent.c
index 4db0e73..4bfa2f9 100644
--- a/net/ipv4/netfilter/ipt_recent.c
+++ b/net/ipv4/netfilter/ipt_recent.c
@@ -23,6 +23,7 @@
#include <linux/bitops.h>
#include <linux/skbuff.h>
#include <linux/inet.h>
+#include <net/net_namespace.h>

#include <linux/netfilter_ipv4/ip_tables.h>
#include <linux/netfilter_ipv4/ipt_recent.h>
@@ -483,7 +484,7 @@ static int __init ipt_recent_init(void)
#ifdef CONFIG_PROC_FS
    if (err)
        return err;
- proc_dir = proc_mkdir("ipt_recent", proc_net);
+ proc_dir = proc_mkdir("ipt_recent", per_net(proc_net, init_net()));
    if (proc_dir == NULL) {
        ipt_unregister_match(&recent_match);
        err = -ENOMEM;
@@ -497,7 +498,7 @@ static void __exit ipt_recent_exit(void)
    BUG_ON(!list_empty(&tables));
    ipt_unregister_match(&recent_match);
#ifdef CONFIG_PROC_FS
- remove_proc_entry("ipt_recent", proc_net);
+ remove_proc_entry("ipt_recent", per_net(proc_net, init_net()));

```

```

#endif
}

diff --git a/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
b/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
index 3b31bc6..ebdb56e 100644
--- a/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
+++ b/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
@@ -11,6 +11,7 @@
#include <linux/proc_fs.h>
#include <linux/seq_file.h>
#include <linux/percpu.h>
+#include <net/net_namespace.h>

#include <linux/netfilter.h>
#include <net/netfilter/nf_conntrack_core.h>
@@ -378,16 +379,16 @@ int __init nf_conntrack_ipv4_compat_init(void)
{
    struct proc_dir_entry *proc, *proc_exp, *proc_stat;

-   proc = proc_net_fops_create("ip_conntrack", 0440, &ct_file_ops);
+   proc = proc_net_fops_create(init_net(), "ip_conntrack", 0440, &ct_file_ops);
    if (!proc)
        goto err1;

-   proc_exp = proc_net_fops_create("ip_conntrack_expect", 0440,
+   proc_exp = proc_net_fops_create(init_net(), "ip_conntrack_expect", 0440,
        &ip_exp_file_ops);
    if (!proc_exp)
        goto err2;

-   proc_stat = create_proc_entry("ip_conntrack", S_IRUGO, proc_net_stat);
+   proc_stat = create_proc_entry("ip_conntrack", S_IRUGO, per_net(proc_net_stat, init_net()));
    if (!proc_stat)
        goto err3;

@@ -397,16 +398,16 @@ int __init nf_conntrack_ipv4_compat_init(void)
    return 0;

err3:
-   proc_net_remove("ip_conntrack_expect");
+   proc_net_remove(init_net(), "ip_conntrack_expect");
err2:
-   proc_net_remove("ip_conntrack");
+   proc_net_remove(init_net(), "ip_conntrack");
err1:
    return -ENOMEM;
}

```

```

void __exit nf_conntrack_ipv4_compat_fini(void)
{
- remove_proc_entry("ip_conntrack", proc_net_stat);
- proc_net_remove("ip_conntrack_expect");
- proc_net_remove("ip_conntrack");
+ remove_proc_entry("ip_conntrack", per_net(proc_net_stat, init_net()));
+ proc_net_remove(init_net(), "ip_conntrack_expect");
+ proc_net_remove(init_net(), "ip_conntrack");
}
diff --git a/net/ipv4/proc.c b/net/ipv4/proc.c
index cd873da..c9c5601 100644
--- a/net/ipv4/proc.c
+++ b/net/ipv4/proc.c
@@ -44,6 +44,7 @@
#include <linux/seq_file.h>
#include <net/sock.h>
#include <net/raw.h>
+#include <net/net_namespace.h>

static int fold_prot_inuse(struct proto *proto)
{
@@ -372,20 +373,20 @@ int __init ip_misc_proc_init(void)
{
    int rc = 0;

- if (!proc_net_fops_create("netstat", S_IRUGO, &netstat_seq_fops))
+ if (!proc_net_fops_create(init_net(), "netstat", S_IRUGO, &netstat_seq_fops))
    goto out_netstat;

- if (!proc_net_fops_create("snmp", S_IRUGO, &snmp_seq_fops))
+ if (!proc_net_fops_create(init_net(), "snmp", S_IRUGO, &snmp_seq_fops))
    goto out_snmp;

- if (!proc_net_fops_create("sockstat", S_IRUGO, &sockstat_seq_fops))
+ if (!proc_net_fops_create(init_net(), "sockstat", S_IRUGO, &sockstat_seq_fops))
    goto out_sockstat;
out:
    return rc;
out_sockstat:
- proc_net_remove("snmp");
+ proc_net_remove(init_net(), "snmp");
out_snmp:
- proc_net_remove("netstat");
+ proc_net_remove(init_net(), "netstat");
out_netstat:
    rc = -ENOMEM;
    goto out;

```

```

diff --git a/net/ipv4/raw.c b/net/ipv4/raw.c
index a6c63bb..38fe668 100644
--- a/net/ipv4/raw.c
+++ b/net/ipv4/raw.c
@@ -73,6 +73,7 @@
#include <net/inet_common.h>
#include <net/checksum.h>
#include <net/xfrm.h>
+#include <net/net_namespace.h>
#include <linux/rtnetlink.h>
#include <linux/proc_fs.h>
#include <linux/seq_file.h>
@@ -926,13 +927,13 @@ static struct file_operations raw_seq_fops = {

int __init raw_proc_init(void)
{
- if (!proc_net_fops_create("raw", S_IRUGO, &raw_seq_fops))
+ if (!proc_net_fops_create(init_net(), "raw", S_IRUGO, &raw_seq_fops))
    return -ENOMEM;
    return 0;
}

void __init raw_proc_exit(void)
{
- proc_net_remove("raw");
+ proc_net_remove(init_net(), "raw");
}
#endif /* CONFIG_PROC_FS */
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index 2daa0dc..8be7506 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -105,6 +105,7 @@
#include <net/xfrm.h>
#include <net/ip_mp_alg.h>
#include <net/netevent.h>
+#include <net/net_namespace.h>
#ifdef CONFIG_SYSCTL
#include <linux/sysctl.h>
#endif
@@ -3178,15 +3179,15 @@ int __init ip_rt_init(void)
#ifdef CONFIG_PROC_FS
{
    struct proc_dir_entry *rtstat_pde = NULL; /* keep gcc happy */
- if (!proc_net_fops_create("rt_cache", S_IRUGO, &rt_cache_seq_fops) ||
+ if (!proc_net_fops_create(init_net(), "rt_cache", S_IRUGO, &rt_cache_seq_fops) ||
    !(rtstat_pde = create_proc_entry("rt_cache", S_IRUGO,
-      proc_net_stat))) {

```

```

+         per_net(proc_net_stat, init_net())) {
    return -ENOMEM;
}
rtstat_pde->proc_fops = &rt_cpu_seq_fops;
}
#ifdef CONFIG_NET_CLS_ROUTE
- create_proc_read_entry("rt_acct", 0, proc_net, ip_rt_acct_read, NULL);
+ create_proc_read_entry("rt_acct", 0, per_net(proc_net, init_net()), ip_rt_acct_read, NULL);
#endif
#endif
#ifdef CONFIG_XFRM
diff --git a/net/ipv4/tcp_ipv4.c b/net/ipv4/tcp_ipv4.c
index 12de90a..ee4306f 100644
--- a/net/ipv4/tcp_ipv4.c
+++ b/net/ipv4/tcp_ipv4.c
@@ -71,6 +71,7 @@
#include <net/timewait_sock.h>
#include <net/xfrm.h>
#include <net/netdma.h>
+#include <net/net_namespace.h>

#include <linux/inet.h>
#include <linux/ipv6.h>
@@ -2252,7 +2253,7 @@ int tcp_proc_register(struct tcp_seq_afinfo *afinfo)
    afinfo->seq_fops->llseek = seq_llseek;
    afinfo->seq_fops->release = seq_release_private;

- p = proc_net_fops_create(afinfo->name, S_IRUGO, afinfo->seq_fops);
+ p = proc_net_fops_create(init_net(), afinfo->name, S_IRUGO, afinfo->seq_fops);
    if (p)
        p->data = afinfo;
    else
@@ -2264,7 +2265,7 @@ void tcp_proc_unregister(struct tcp_seq_afinfo *afinfo)
{
    if (!afinfo)
        return;
- proc_net_remove(afinfo->name);
+ proc_net_remove(init_net(), afinfo->name);
    memset(afinfo->seq_fops, 0, sizeof(*afinfo->seq_fops));
}

diff --git a/net/ipv4/tcp_probe.c b/net/ipv4/tcp_probe.c
index f230eee..e8a3d96 100644
--- a/net/ipv4/tcp_probe.c
+++ b/net/ipv4/tcp_probe.c
@@ -159,7 +159,7 @@ static __init int tcpprobe_init(void)
    if (IS_ERR(tcpw.fifo))
        return PTR_ERR(tcpw.fifo);

```

```

- if (!proc_net_fops_create(procname, S_IRUSR, &tcpprobe_fops))
+ if (!proc_net_fops_create(init_net(), procname, S_IRUSR, &tcpprobe_fops))
    goto err0;

    ret = register_jprobe(&tcp_send_probe);
@@ -169,7 +169,7 @@ static __init int tcpprobe_init(void)
    pr_info("TCP watch registered (port=%d)\n", port);
    return 0;
err1:
- proc_net_remove(procname);
+ proc_net_remove(init_net(), procname);
err0:
    kfifo_free(tcpw.fifo);
    return ret;
@@ -179,7 +179,7 @@ module_init(tcpprobe_init);
static __exit void tcpprobe_exit(void)
{
    kfifo_free(tcpw.fifo);
- proc_net_remove(procname);
+ proc_net_remove(init_net(), procname);
    unregister_jprobe(&tcp_send_probe);

}
diff --git a/net/ipv4/udp.c b/net/ipv4/udp.c
index cfff930..7527183 100644
--- a/net/ipv4/udp.c
+++ b/net/ipv4/udp.c
@@ -101,6 +101,7 @@
#include <net/route.h>
#include <net/checksum.h>
#include <net/xfrm.h>
+#include <net/net_namespace.h>
#include "udp_impl.h"

/*
@@ -1643,7 +1644,7 @@ int udp_proc_register(struct udp_seq_afinfo *afinfo)
    afinfo->seq_fops->llseek = seq_llseek;
    afinfo->seq_fops->release = seq_release_private;

- p = proc_net_fops_create(afinfo->name, S_IRUGO, afinfo->seq_fops);
+ p = proc_net_fops_create(init_net(), afinfo->name, S_IRUGO, afinfo->seq_fops);
    if (p)
        p->data = afinfo;
    else
@@ -1655,7 +1656,7 @@ void udp_proc_unregister(struct udp_seq_afinfo *afinfo)
{
    if (!afinfo)

```

```

    return;
- proc_net_remove(ainfo->name);
+ proc_net_remove(init_net(), ainfo->name);
    memset(ainfo->seq_fops, 0, sizeof(*ainfo->seq_fops));
}

```

```

diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 6aded83..52bd4dd 100644

```

```

--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -74,6 +74,7 @@
#include <net/tcp.h>
#include <net/ip.h>
#include <net/netlink.h>
+#include <net/net_namespace.h>
#include <linux/if_tunnel.h>
#include <linux/rtnetlink.h>

```

```

@@ -2780,14 +2781,14 @@ static struct file_operations if6_fops = {

```

```

int __init if6_proc_init(void)
{
- if (!proc_net_fops_create("if_inet6", S_IRUGO, &if6_fops))
+ if (!proc_net_fops_create(init_net(), "if_inet6", S_IRUGO, &if6_fops))
    return -ENOMEM;
    return 0;
}

```

```

void if6_proc_exit(void)
{
- proc_net_remove("if_inet6");
+ proc_net_remove(init_net(), "if_inet6");
}
#endif /* CONFIG_PROC_FS */

```

```

@@ -4143,6 +4144,6 @@ void __exit addrconf_cleanup(void)
    rtnl_unlock();

```

```

#ifdef CONFIG_PROC_FS
- proc_net_remove("if_inet6");
+ proc_net_remove(init_net(), "if_inet6");
#endif
}

```

```

diff --git a/net/ipv6/anycast.c b/net/ipv6/anycast.c
index a960476..c42bad9 100644

```

```

--- a/net/ipv6/anycast.c
+++ b/net/ipv6/anycast.c
@@ -33,6 +33,7 @@

```

```

#include <net/sock.h>
#include <net/snmp.h>
+#include <net/net_namespace.h>

#include <net/ipv6.h>
#include <net/protocol.h>
@@ -575,7 +576,7 @@ static struct file_operations ac6_seq_fops = {

int __init ac6_proc_init(void)
{
- if (!proc_net_fops_create("anycast6", S_IRUGO, &ac6_seq_fops))
+ if (!proc_net_fops_create(init_net(), "anycast6", S_IRUGO, &ac6_seq_fops))
    return -ENOMEM;

    return 0;
@@ -583,7 +584,7 @@ int __init ac6_proc_init(void)

void ac6_proc_exit(void)
{
- proc_net_remove("anycast6");
+ proc_net_remove(init_net(), "anycast6");
}
#endif

diff --git a/net/ipv6/ip6_flowlabel.c b/net/ipv6/ip6_flowlabel.c
index 624fae2..350aedb 100644
--- a/net/ipv6/ip6_flowlabel.c
+++ b/net/ipv6/ip6_flowlabel.c
@@ -22,6 +22,7 @@
#include <linux/seq_file.h>

#include <net/sock.h>
+#include <net/net_namespace.h>

#include <net/ipv6.h>
#include <net/ndisc.h>
@@ -690,7 +691,7 @@ static struct file_operations ip6fl_seq_fops = {
void ip6_flowlabel_init(void)
{
#ifdef CONFIG_PROC_FS
- proc_net_fops_create("ip6_flowlabel", S_IRUGO, &ip6fl_seq_fops);
+ proc_net_fops_create(init_net(), "ip6_flowlabel", S_IRUGO, &ip6fl_seq_fops);
#endif
}

@@ -698,6 +699,6 @@ void ip6_flowlabel_cleanup(void)
{

```

```

    del_timer(&ip6_fl_gc_timer);
#ifdef CONFIG_PROC_FS
- proc_net_remove("ip6_flowlabel");
+ proc_net_remove(init_net(), "ip6_flowlabel");
#endif
}
diff --git a/net/ipv6/mcast.c b/net/ipv6/mcast.c
index a1c231a..2759571 100644
--- a/net/ipv6/mcast.c
+++ b/net/ipv6/mcast.c
@@ -51,6 +51,7 @@

#include <net/sock.h>
#include <net/snmp.h>
+#include <net/net_namespace.h>

#include <net/ipv6.h>
#include <net/protocol.h>
@@ -2661,8 +2662,8 @@ int __init igmp6_init(struct net_proto_family *ops)
    np->hop_limit = 1;

#ifdef CONFIG_PROC_FS
- proc_net_fops_create("igmp6", S_IRUGO, &igmp6_mc_seq_fops);
- proc_net_fops_create("mcfilter6", S_IRUGO, &igmp6_mcf_seq_fops);
+ proc_net_fops_create(init_net(), "igmp6", S_IRUGO, &igmp6_mc_seq_fops);
+ proc_net_fops_create(init_net(), "mcfilter6", S_IRUGO, &igmp6_mcf_seq_fops);
#endif

    return 0;
@@ -2674,7 +2675,7 @@ void igmp6_cleanup(void)
    igmp6_socket = NULL; /* for safety */

#ifdef CONFIG_PROC_FS
- proc_net_remove("mcfilter6");
- proc_net_remove("igmp6");
+ proc_net_remove(init_net(), "mcfilter6");
+ proc_net_remove(init_net(), "igmp6");
#endif
}
diff --git a/net/ipv6/netfilter/ip6_queue.c b/net/ipv6/netfilter/ip6_queue.c
index e774be7..45b64a5 100644
--- a/net/ipv6/netfilter/ip6_queue.c
+++ b/net/ipv6/netfilter/ip6_queue.c
@@ -36,6 +36,7 @@

#include <linux/sysctl.h>
#include <linux/proc_fs.h>
#include <linux/mutex.h>
+#include <net/net_namespace.h>

```

```

#include <net/sock.h>
#include <net/ipv6.h>
#include <net/ip6_route.h>
@@ -674,7 +675,7 @@ static int __init ip6_queue_init(void)
    goto cleanup_netlink_notifier;
}

- proc = proc_net_create(IPQ_PROC_FS_NAME, 0, ipq_get_info);
+ proc = proc_net_create(init_net(), IPQ_PROC_FS_NAME, 0, ipq_get_info);
    if (proc)
        proc->owner = THIS_MODULE;
    else {
@@ -695,7 +696,7 @@ static int __init ip6_queue_init(void)
cleanup_sysctl:
    unregister_sysctl_table(ipq_sysctl_header);
    unregister_netdevice_notifier(&ipq_dev_notifier);
- proc_net_remove(IPQ_PROC_FS_NAME);
+ proc_net_remove(init_net(), IPQ_PROC_FS_NAME);

cleanup_ipqnl:
    sock_release(ipqnl->sk_socket);
@@ -715,7 +716,7 @@ static void __exit ip6_queue_fini(void)

    unregister_sysctl_table(ipq_sysctl_header);
    unregister_netdevice_notifier(&ipq_dev_notifier);
- proc_net_remove(IPQ_PROC_FS_NAME);
+ proc_net_remove(init_net(), IPQ_PROC_FS_NAME);

    sock_release(ipqnl->sk_socket);
    mutex_lock(&ipqnl_mutex);
diff --git a/net/ipv6/proc.c b/net/ipv6/proc.c
index 35249d8..1827885 100644
--- a/net/ipv6/proc.c
+++ b/net/ipv6/proc.c
@@ -28,6 +28,7 @@
#include <net/tcp.h>
#include <net/transp_v6.h>
#include <net/ipv6.h>
+#include <net/net_namespace.h>

#ifdef CONFIG_PROC_FS
static struct proc_dir_entry *proc_net_devsnmp6;
@@ -244,22 +245,22 @@ int __init ipv6_misc_proc_init(void)
{
    int rc = 0;

- if (!proc_net_fops_create("snmp6", S_IRUGO, &snmp6_seq_fops))
+ if (!proc_net_fops_create(init_net(), "snmp6", S_IRUGO, &snmp6_seq_fops))

```

```

goto proc_snmp6_fail;

- proc_net_devsnmp6 = proc_mkdir("dev_snmp6", proc_net);
+ proc_net_devsnmp6 = proc_mkdir("dev_snmp6", per_net(proc_net, init_net()));
  if (!proc_net_devsnmp6)
    goto proc_dev_snmp6_fail;

- if (!proc_net_fops_create("sockstat6", S_IRUGO, &sockstat6_seq_fops))
+ if (!proc_net_fops_create(init_net(), "sockstat6", S_IRUGO, &sockstat6_seq_fops))
  goto proc_sockstat6_fail;
out:
return rc;

proc_sockstat6_fail:
- proc_net_remove("dev_snmp6");
+ proc_net_remove(init_net(), "dev_snmp6");
proc_dev_snmp6_fail:
- proc_net_remove("snmp6");
+ proc_net_remove(init_net(), "snmp6");
proc_snmp6_fail:
  rc = -ENOMEM;
  goto out;
@@ -267,9 +268,9 @@ proc_snmp6_fail:

void ipv6_misc_proc_exit(void)
{
- proc_net_remove("sockstat6");
- proc_net_remove("dev_snmp6");
- proc_net_remove("snmp6");
+ proc_net_remove(init_net(), "sockstat6");
+ proc_net_remove(init_net(), "dev_snmp6");
+ proc_net_remove(init_net(), "snmp6");
}

#else /* CONFIG_PROC_FS */
diff --git a/net/ipv6/raw.c b/net/ipv6/raw.c
index 4ae1b19..2e1825c 100644
--- a/net/ipv6/raw.c
+++ b/net/ipv6/raw.c
@@ -50,6 +50,7 @@
#include <net/udp.h>
#include <net/inet_common.h>
#include <net/tcp_states.h>
+#include <net/net_namespace.h>
#ifdef CONFIG_IPV6_MIP6
#include <net/mip6.h>
#endif
@@ -1274,13 +1275,13 @@ static struct file_operations raw6_seq_fops = {

```

```

int __init raw6_proc_init(void)
{
- if (!proc_net_fops_create("raw6", S_IRUGO, &raw6_seq_fops))
+ if (!proc_net_fops_create(init_net(), "raw6", S_IRUGO, &raw6_seq_fops))
    return -ENOMEM;
    return 0;
}

void raw6_proc_exit(void)
{
- proc_net_remove("raw6");
+ proc_net_remove(init_net(), "raw6");
}
#endif /* CONFIG_PROC_FS */
diff --git a/net/ipv6/route.c b/net/ipv6/route.c
index 8c3d568..8c9fef9 100644
--- a/net/ipv6/route.c
+++ b/net/ipv6/route.c
@@ -56,6 +56,7 @@
#include <net/xfrm.h>
#include <net/netevent.h>
#include <net/netlink.h>
+#include <net/net_namespace.h>

#include <asm/uaccess.h>

@@ -2458,11 +2459,11 @@ void __init ip6_route_init(void)
    SLAB_HWCACHE_ALIGN|SLAB_PANIC, NULL, NULL);
    fib6_init();
#ifdef CONFIG_PROC_FS
- p = proc_net_create("ipv6_route", 0, rt6_proc_info);
+ p = proc_net_create(init_net(), "ipv6_route", 0, rt6_proc_info);
    if (p)
        p->owner = THIS_MODULE;

- proc_net_fops_create("rt6_stats", S_IRUGO, &rt6_stats_seq_fops);
+ proc_net_fops_create(init_net(), "rt6_stats", S_IRUGO, &rt6_stats_seq_fops);
#endif
#ifdef CONFIG_XFRM
    xfrm6_init();
@@ -2478,8 +2479,8 @@ void ip6_route_cleanup(void)
    fib6_rules_cleanup();
#endif
#ifdef CONFIG_PROC_FS
- proc_net_remove("ipv6_route");
- proc_net_remove("rt6_stats");
+ proc_net_remove(init_net(), "ipv6_route");

```

```

+ proc_net_remove(init_net(), "rt6_stats");
#endif
#ifdef CONFIG_XFRM
    xfrm6_fini();
diff --git a/net/ipx/ipx_proc.c b/net/ipx/ipx_proc.c
index b7463df..bda8775 100644
--- a/net/ipx/ipx_proc.c
+++ b/net/ipx/ipx_proc.c
@@ -9,6 +9,7 @@
#include <linux/proc_fs.h>
#include <linux/spinlock.h>
#include <linux/seq_file.h>
+#include <net/net_namespace.h>
#include <net/tcp_states.h>
#include <net/ipx.h>

@@ -353,7 +354,7 @@ int __init ipx_proc_init(void)
    struct proc_dir_entry *p;
    int rc = -ENOMEM;

- ipx_proc_dir = proc_mkdir("ipx", proc_net);
+ ipx_proc_dir = proc_mkdir("ipx", per_net(proc_net, init_net()));

    if (!ipx_proc_dir)
        goto out;
@@ -381,7 +382,7 @@ out_socket:
out_route:
    remove_proc_entry("interface", ipx_proc_dir);
out_interface:
- remove_proc_entry("ipx", proc_net);
+ remove_proc_entry("ipx", per_net(proc_net, init_net()));
    goto out;
}

@@ -390,7 +391,7 @@ void __exit ipx_proc_exit(void)
    remove_proc_entry("interface", ipx_proc_dir);
    remove_proc_entry("route", ipx_proc_dir);
    remove_proc_entry("socket", ipx_proc_dir);
- remove_proc_entry("ipx", proc_net);
+ remove_proc_entry("ipx", per_net(proc_net, init_net()));
}

#else /* CONFIG_PROC_FS */
diff --git a/net/irda/irproc.c b/net/irda/irproc.c
index 88b9c43..0af0f55 100644
--- a/net/irda/irproc.c
+++ b/net/irda/irproc.c
@@ -28,6 +28,7 @@

```

```

#include <linux/seq_file.h>
#include <linux/module.h>
#include <linux/init.h>
+#include <net/net_namespace.h>

#include <net/irda/irda.h>
#include <net/irda/irlap.h>
@@ -66,7 +67,7 @@ void __init irda_proc_register(void)
    int i;
    struct proc_dir_entry *d;

- proc_irda = proc_mkdir("irda", proc_net);
+ proc_irda = proc_mkdir("irda", per_net(proc_net, init_net()));
    if (proc_irda == NULL)
        return;
    proc_irda->owner = THIS_MODULE;
@@ -92,7 +93,7 @@ void __exit irda_proc_unregister(void)
    for (i=0; i<ARRAY_SIZE(irda_dirs); i++)
        remove_proc_entry(irda_dirs[i].name, proc_irda);

-     remove_proc_entry("irda", proc_net);
+     remove_proc_entry("irda", per_net(proc_net, init_net()));
    proc_irda = NULL;
}
}
diff --git a/net/key/af_key.c b/net/key/af_key.c
index 5dd5094..c79f9c4 100644
--- a/net/key/af_key.c
+++ b/net/key/af_key.c
@@ -28,6 +28,7 @@
#include <linux/init.h>
#include <net/xfrm.h>
#include <linux/audit.h>
+#include <net/net_namespace.h>

#include <net/sock.h>

@@ -3292,7 +3293,7 @@ static struct xfrm_mgr pfkeyv2_mgr =
static void __exit ipsec_pfkey_exit(void)
{
    xfrm_unregister_km(&pfkeyv2_mgr);
- remove_proc_entry("net/pfkey", NULL);
+ remove_proc_entry("pfkey", per_net(proc_net, init_net()));
    sock_unregister(PF_KEY);
    proto_unregister(&key_proto);
}
@@ -3309,7 +3310,7 @@ static int __init ipsec_pfkey_init(void)
    goto out_unregister_key_proto;

```

```

#ifdef CONFIG_PROC_FS
    err = -ENOMEM;
- if (create_proc_read_entry("net/pfkey", 0, NULL, pfkey_read_proc, NULL) == NULL)
+ if (create_proc_read_entry("pfkey", 0, per_net(proc_net, init_net()), pfkey_read_proc, NULL) ==
NULL)
    goto out_sock_unregister;
#endif
    err = xfrm_register_km(&pfkeyv2_mgr);
diff --git a/net/llc/llc_proc.c b/net/llc/llc_proc.c
index 19308fe..4d0a804 100644
--- a/net/llc/llc_proc.c
+++ b/net/llc/llc_proc.c
@@ -18,6 +18,7 @@
#include <linux/errno.h>
#include <linux/seq_file.h>
#include <net/sock.h>
+#include <net/net_namespace.h>
#include <net/llc.h>
#include <net/llc_c_ac.h>
#include <net/llc_c_ev.h>
@@ -231,7 +232,7 @@ int __init llc_proc_init(void)
    int rc = -ENOMEM;
    struct proc_dir_entry *p;

- llc_proc_dir = proc_mkdir("llc", proc_net);
+ llc_proc_dir = proc_mkdir("llc", per_net(proc_net, init_net()));
    if (!llc_proc_dir)
        goto out;
    llc_proc_dir->owner = THIS_MODULE;
@@ -254,7 +255,7 @@ out:
out_core:
    remove_proc_entry("socket", llc_proc_dir);
out_socket:
- remove_proc_entry("llc", proc_net);
+ remove_proc_entry("llc", per_net(proc_net, init_net()));
    goto out;
}

@@ -262,5 +263,5 @@ void llc_proc_exit(void)
{
    remove_proc_entry("socket", llc_proc_dir);
    remove_proc_entry("core", llc_proc_dir);
- remove_proc_entry("llc", proc_net);
+ remove_proc_entry("llc", per_net(proc_net, init_net()));
}
diff --git a/net/netfilter/core.c b/net/netfilter/core.c
index 291b8c6..cafa00c 100644
--- a/net/netfilter/core.c

```

```

+++ b/net/netfilter/core.c
@@ -23,6 +23,7 @@
#include <linux/inetdevice.h>
#include <linux/proc_fs.h>
#include <net/sock.h>
+#include <net/net_namespace.h>

#include "nf_internals.h"

@@ -269,7 +270,7 @@ void __init netfilter_init(void)
}

#ifdef CONFIG_PROC_FS
- proc_net_netfilter = proc_mkdir("netfilter", proc_net);
+ proc_net_netfilter = proc_mkdir("netfilter", per_net(proc_net, init_net()));
if (!proc_net_netfilter)
panic("cannot create netfilter proc entry");
#endif
diff --git a/net/netfilter/nf_conntrack_standalone.c b/net/netfilter/nf_conntrack_standalone.c
index 2587b49..314dc2c 100644
--- a/net/netfilter/nf_conntrack_standalone.c
+++ b/net/netfilter/nf_conntrack_standalone.c
@@ -25,6 +25,7 @@
#include <linux/seq_file.h>
#include <linux/percpu.h>
#include <linux/netdevice.h>
+#include <net/net_namespace.h>
#ifdef CONFIG_SYSCTL
#include <linux/sysctl.h>
#endif
@@ -430,14 +431,14 @@ static int __init nf_conntrack_standalone_init(void)
return ret;

#ifdef CONFIG_PROC_FS
- proc = proc_net_fops_create("nf_conntrack", 0440, &ct_file_ops);
+ proc = proc_net_fops_create(init_net(), "nf_conntrack", 0440, &ct_file_ops);
if (!proc) goto cleanup_init;

- proc_exp = proc_net_fops_create("nf_conntrack_expect", 0440,
+ proc_exp = proc_net_fops_create(init_net(), "nf_conntrack_expect", 0440,
&exp_file_ops);
if (!proc_exp) goto cleanup_proc;

- proc_stat = create_proc_entry("nf_conntrack", S_IRUGO, proc_net_stat);
+ proc_stat = create_proc_entry("nf_conntrack", S_IRUGO, per_net(proc_net_stat, init_net()));
if (!proc_stat)
goto cleanup_proc_exp;

```

```

@@ -458,11 +459,11 @@ static int __init nf_conntrack_standalone_init(void)
    cleanup_proc_stat:
#endif
#ifdef CONFIG_PROC_FS
- remove_proc_entry("nf_conntrack", proc_net_stat);
+ remove_proc_entry("nf_conntrack", per_net(proc_net_stat, init_net()));
    cleanup_proc_exp:
- proc_net_remove("nf_conntrack_expect");
+ proc_net_remove(init_net(), "nf_conntrack_expect");
    cleanup_proc:
- proc_net_remove("nf_conntrack");
+ proc_net_remove(init_net(), "nf_conntrack");
    cleanup_init:
#endif /* CONFIG_PROC_FS */
    nf_conntrack_cleanup();
@@ -475,9 +476,9 @@ static void __exit nf_conntrack_standalone_fini(void)
    unregister_sysctl_table(nf_ct_sysctl_header);
#endif
#ifdef CONFIG_PROC_FS
- remove_proc_entry("nf_conntrack", proc_net_stat);
- proc_net_remove("nf_conntrack_expect");
- proc_net_remove("nf_conntrack");
+ remove_proc_entry("nf_conntrack", per_net(proc_net_stat, init_net()));
+ proc_net_remove(init_net(), "nf_conntrack_expect");
+ proc_net_remove(init_net(), "nf_conntrack");
#endif /* CONFIG_PROC_FS */
    nf_conntrack_cleanup();
}
diff --git a/net/netfilter/x_tables.c b/net/netfilter/x_tables.c
index 8996584..9fb3491 100644
--- a/net/netfilter/x_tables.c
+++ b/net/netfilter/x_tables.c
@@ -22,6 +22,7 @@
#include <linux/vmalloc.h>
#include <linux/mutex.h>
#include <linux/mm.h>
+#include <net/net_namespace.h>

#include <linux/netfilter/x_tables.h>
#include <linux/netfilter_arp.h>
@@ -800,7 +801,7 @@ int xt_proto_init(int af)
#ifdef CONFIG_PROC_FS
    strcpy(buf, xt_proto_prefix[af], sizeof(buf));
    strlcat(buf, FORMAT_TABLES, sizeof(buf));
- proc = proc_net_fops_create(buf, 0440, &xt_file_ops);
+ proc = proc_net_fops_create(init_net(), buf, 0440, &xt_file_ops);
    if (!proc)
        goto out;

```

```

proc->data = (void *) ((unsigned long) af | (TABLE << 16));
@@ -808,14 +809,14 @@ int xt_proto_init(int af)

    strcpy(buf, xt_proto_prefix[af], sizeof(buf));
    strcat(buf, FORMAT_MATCHES, sizeof(buf));
- proc = proc_net_fops_create(buf, 0440, &xt_file_ops);
+ proc = proc_net_fops_create(init_net(), buf, 0440, &xt_file_ops);
    if (!proc)
        goto out_remove_tables;
    proc->data = (void *) ((unsigned long) af | (MATCH << 16));

    strcpy(buf, xt_proto_prefix[af], sizeof(buf));
    strcat(buf, FORMAT_TARGETS, sizeof(buf));
- proc = proc_net_fops_create(buf, 0440, &xt_file_ops);
+ proc = proc_net_fops_create(init_net(), buf, 0440, &xt_file_ops);
    if (!proc)
        goto out_remove_matches;
    proc->data = (void *) ((unsigned long) af | (TARGET << 16));
@@ -827,12 +828,12 @@ int xt_proto_init(int af)
out_remove_matches:
    strcpy(buf, xt_proto_prefix[af], sizeof(buf));
    strcat(buf, FORMAT_MATCHES, sizeof(buf));
- proc_net_remove(buf);
+ proc_net_remove(init_net(), buf);

out_remove_tables:
    strcpy(buf, xt_proto_prefix[af], sizeof(buf));
    strcat(buf, FORMAT_TABLES, sizeof(buf));
- proc_net_remove(buf);
+ proc_net_remove(init_net(), buf);
out:
    return -1;
#endif
@@ -846,15 +847,15 @@ void xt_proto_fini(int af)

    strcpy(buf, xt_proto_prefix[af], sizeof(buf));
    strcat(buf, FORMAT_TABLES, sizeof(buf));
- proc_net_remove(buf);
+ proc_net_remove(init_net(), buf);

    strcpy(buf, xt_proto_prefix[af], sizeof(buf));
    strcat(buf, FORMAT_TARGETS, sizeof(buf));
- proc_net_remove(buf);
+ proc_net_remove(init_net(), buf);

    strcpy(buf, xt_proto_prefix[af], sizeof(buf));
    strcat(buf, FORMAT_MATCHES, sizeof(buf));
- proc_net_remove(buf);

```

```

+ proc_net_remove(init_net(), buf);
#endif /*CONFIG_PROC_FS*/
}
EXPORT_SYMBOL_GPL(xt_proto_fini);
diff --git a/net/netfilter/xt_hashlimit.c b/net/netfilter/xt_hashlimit.c
index f28bf69..21c07df 100644
--- a/net/netfilter/xt_hashlimit.c
+++ b/net/netfilter/xt_hashlimit.c
@@ -21,6 +21,7 @@
#include <linux/in.h>
#include <linux/ip.h>
#include <linux/ipv6.h>
+#include <net/net_namespace.h>

#include <linux/netfilter/x_tables.h>
#include <linux/netfilter_ipv4/ip_tables.h>
@@ -737,13 +738,13 @@ static int __init xt_hashlimit_init(void)
    printk(KERN_ERR "xt_hashlimit: unable to create slab cache\n");
    goto err2;
}
- hashlimit_procd4 = proc_mkdir("ipt_hashlimit", proc_net);
+ hashlimit_procd4 = proc_mkdir("ipt_hashlimit", per_net(proc_net, init_net()));
if (!hashlimit_procd4) {
    printk(KERN_ERR "xt_hashlimit: unable to create proc dir "
        "entry\n");
    goto err3;
}
- hashlimit_procd6 = proc_mkdir("ip6t_hashlimit", proc_net);
+ hashlimit_procd6 = proc_mkdir("ip6t_hashlimit", per_net(proc_net, init_net()));
if (!hashlimit_procd6) {
    printk(KERN_ERR "xt_hashlimit: unable to create proc dir "
        "entry\n");
@@ -751,7 +752,7 @@ static int __init xt_hashlimit_init(void)
}
return 0;
err4:
- remove_proc_entry("ipt_hashlimit", proc_net);
+ remove_proc_entry("ipt_hashlimit", per_net(proc_net, init_net()));
err3:
    kmem_cache_destroy(hashlimit_cachep);
err2:
@@ -763,8 +764,8 @@ err1:

static void __exit xt_hashlimit_fini(void)
{
- remove_proc_entry("ipt_hashlimit", proc_net);
- remove_proc_entry("ip6t_hashlimit", proc_net);
+ remove_proc_entry("ipt_hashlimit", per_net(proc_net, init_net()));

```

```

+ remove_proc_entry("ip6t_hashlimit", per_net(proc_net, init_net()));
  kmem_cache_destroy(hashlimit_cachep);
  xt_unregister_matches(xt_hashlimit, ARRAY_SIZE(xt_hashlimit));
}
diff --git a/net/netlink/af_netlink.c b/net/netlink/af_netlink.c
index 383dd4e..3c00f48 100644
--- a/net/netlink/af_netlink.c
+++ b/net/netlink/af_netlink.c
@@ -60,6 +60,7 @@
#include <net/sock.h>
#include <net/scm.h>
#include <net/netlink.h>
+#include <net/net_namespace.h>

#define NLGRPSZ(x) (ALIGN(x, sizeof(unsigned long) * 8) / 8)

@@ -1806,7 +1807,7 @@ static int __init netlink_proto_init(void)

  sock_register(&netlink_family_ops);
#ifdef CONFIG_PROC_FS
- proc_net_fops_create("netlink", 0, &netlink_seq_fops);
+ proc_net_fops_create(init_net(), "netlink", 0, &netlink_seq_fops);
#endif
/* The netlink device handler may be needed early. */
rtnetlink_init();
diff --git a/net/netrom/af_netrom.c b/net/netrom/af_netrom.c
index 43bbe2c..601d58c 100644
--- a/net/netrom/af_netrom.c
+++ b/net/netrom/af_netrom.c
@@ -41,6 +41,7 @@
#include <net/ip.h>
#include <net/tcp_states.h>
#include <net/arp.h>
+#include <net/net_namespace.h>
#include <linux/init.h>

static int nr_ndevs = 4;
@@ -1442,9 +1443,9 @@ static int __init nr_proto_init(void)

  nr_loopback_init();

- proc_net_fops_create("nr", S_IRUGO, &nr_info_fops);
- proc_net_fops_create("nr_neigh", S_IRUGO, &nr_neigh_fops);
- proc_net_fops_create("nr_nodes", S_IRUGO, &nr_nodes_fops);
+ proc_net_fops_create(init_net(), "nr", S_IRUGO, &nr_info_fops);
+ proc_net_fops_create(init_net(), "nr_neigh", S_IRUGO, &nr_neigh_fops);
+ proc_net_fops_create(init_net(), "nr_nodes", S_IRUGO, &nr_nodes_fops);
out:

```

```

    return rc;
fail:
@@ -1472,9 +1473,9 @@ static void __exit nr_exit(void)
{
    int i;

- proc_net_remove("nr");
- proc_net_remove("nr_neigh");
- proc_net_remove("nr_nodes");
+ proc_net_remove(init_net(), "nr");
+ proc_net_remove(init_net(), "nr_neigh");
+ proc_net_remove(init_net(), "nr_nodes");
    nr_loopback_clear();

    nr_rt_free();
diff --git a/net/packet/af_packet.c b/net/packet/af_packet.c
index da73e8a..04e295a 100644
--- a/net/packet/af_packet.c
+++ b/net/packet/af_packet.c
@@ -65,6 +65,7 @@
#include <net/protocol.h>
#include <linux/skbuff.h>
#include <net/sock.h>
+#include <net/net_namespace.h>
#include <linux/errno.h>
#include <linux/timer.h>
#include <asm/system.h>
@@ -1911,7 +1912,7 @@ static struct file_operations packet_seq_fops = {

static void __exit packet_exit(void)
{
- proc_net_remove("packet");
+ proc_net_remove(init_net(), "packet");
    unregister_netdevice_notifier(&packet_netdev_notifier);
    sock_unregister(PF_PACKET);
    proto_unregister(&packet_proto);
@@ -1926,7 +1927,7 @@ static int __init packet_init(void)

    sock_register(&packet_family_ops);
    register_netdevice_notifier(&packet_netdev_notifier);
- proc_net_fops_create("packet", 0, &packet_seq_fops);
+ proc_net_fops_create(init_net(), "packet", 0, &packet_seq_fops);
out:
    return rc;
}
diff --git a/net/rose/af_rose.c b/net/rose/af_rose.c
index 9e27946..5532340 100644
--- a/net/rose/af_rose.c

```

```

+++ b/net/rose/af_rose.c
@@ -45,6 +45,7 @@
#include <net/tcp_states.h>
#include <net/ip.h>
#include <net/arp.h>
+#include <net/net_namespace.h>

static int rose_ndevs = 10;

@@ -1550,10 +1551,10 @@ static int __init rose_proto_init(void)

    rose_add_loopback_neigh();

- proc_net_fops_create("rose", S_IRUGO, &rose_info_fops);
- proc_net_fops_create("rose_neigh", S_IRUGO, &rose_neigh_fops);
- proc_net_fops_create("rose_nodes", S_IRUGO, &rose_nodes_fops);
- proc_net_fops_create("rose_routes", S_IRUGO, &rose_routes_fops);
+ proc_net_fops_create(init_net(), "rose", S_IRUGO, &rose_info_fops);
+ proc_net_fops_create(init_net(), "rose_neigh", S_IRUGO, &rose_neigh_fops);
+ proc_net_fops_create(init_net(), "rose_nodes", S_IRUGO, &rose_nodes_fops);
+ proc_net_fops_create(init_net(), "rose_routes", S_IRUGO, &rose_routes_fops);
    out:
    return rc;
fail:
@@ -1580,10 +1581,10 @@ static void __exit rose_exit(void)
{
    int i;

- proc_net_remove("rose");
- proc_net_remove("rose_neigh");
- proc_net_remove("rose_nodes");
- proc_net_remove("rose_routes");
+ proc_net_remove(init_net(), "rose");
+ proc_net_remove(init_net(), "rose_neigh");
+ proc_net_remove(init_net(), "rose_nodes");
+ proc_net_remove(init_net(), "rose_routes");
    rose_loopback_clear();

    rose_rt_free();
diff --git a/net/rxrpc/proc.c b/net/rxrpc/proc.c
index 29975d9..e7bd87b 100644
--- a/net/rxrpc/proc.c
+++ b/net/rxrpc/proc.c
@@ -14,6 +14,7 @@
#include <linux/module.h>
#include <linux/proc_fs.h>
#include <linux/seq_file.h>
+#include <net/net_namespace.h>

```

```

#include <rxrpc/rxrpc.h>
#include <rxrpc/transport.h>
#include <rxrpc/peer.h>
@@ -133,7 +134,7 @@ int rxrpc_proc_init(void)
{
    struct proc_dir_entry *p;

- proc_rxrpc = proc_mkdir("rxrpc", proc_net);
+ proc_rxrpc = proc_mkdir("rxrpc", per_net(proc_net, init_net()));
    if (!proc_rxrpc)
        goto error;
    proc_rxrpc->owner = THIS_MODULE;
@@ -169,7 +170,7 @@ int rxrpc_proc_init(void)
    error_calls:
    remove_proc_entry("calls", proc_rxrpc);
    error_proc:
- remove_proc_entry("rxrpc", proc_net);
+ remove_proc_entry("rxrpc", per_net(proc_net, init_net()));
    error:
    return -ENOMEM;
} /* end rxrpc_proc_init() */
@@ -185,7 +186,7 @@ void rxrpc_proc_cleanup(void)
    remove_proc_entry("connections", proc_rxrpc);
    remove_proc_entry("calls", proc_rxrpc);

- remove_proc_entry("rxrpc", proc_net);
+ remove_proc_entry("rxrpc", per_net(proc_net, init_net()));

} /* end rxrpc_proc_cleanup() */

diff --git a/net/sched/sch_api.c b/net/sched/sch_api.c
index 65825f4..da7e1eb 100644
--- a/net/sched/sch_api.c
+++ b/net/sched/sch_api.c
@@ -36,6 +36,7 @@
#include <linux/list.h>
#include <linux/bitops.h>

+#include <net/net_namespace.h>
#include <net/sock.h>
#include <net/pkt_sched.h>

@@ -1296,7 +1297,7 @@ static int __init pktsched_init(void)

    register_qdisc(&pfifo_qdisc_ops);
    register_qdisc(&bfifo_qdisc_ops);
- proc_net_fops_create("psched", 0, &psched_fops);
+ proc_net_fops_create(init_net(), "psched", 0, &psched_fops);

```

```

    return 0;
}
diff --git a/net/sctp/protocol.c b/net/sctp/protocol.c
index 225f39b..ea94951 100644
--- a/net/sctp/protocol.c
+++ b/net/sctp/protocol.c
@@ -59,6 +59,7 @@
#include <net/addrconf.h>
#include <net/inet_common.h>
#include <net/inet_ecn.h>
+#include <net/net_namespace.h>

/* Global data structures. */
struct sctp_globals sctp_globals __read_mostly;
@@ -93,7 +94,7 @@ static __init int sctp_proc_init(void)
{
    if (!proc_net_sctp) {
        struct proc_dir_entry *ent;
-       ent = proc_mkdir("net/sctp", NULL);
+       ent = proc_mkdir("sctp", per_net(proc_net, init_net()));
        if (ent) {
            ent->owner = THIS_MODULE;
            proc_net_sctp = ent;
@@ -126,7 +127,7 @@ static void sctp_proc_exit(void)

    if (proc_net_sctp) {
        proc_net_sctp = NULL;
-       remove_proc_entry("net/sctp", NULL);
+       remove_proc_entry("sctp", per_net(proc_net, init_net()));
    }
}

diff --git a/net/sunrpc/stats.c b/net/sunrpc/stats.c
index bd98124..996b71c 100644
--- a/net/sunrpc/stats.c
+++ b/net/sunrpc/stats.c
@@ -22,6 +22,7 @@
#include <linux/sunrpc/clnt.h>
#include <linux/sunrpc/svcsock.h>
#include <linux/sunrpc/metrics.h>
+#include <net/net_namespace.h>

#define RPCDBG_FACILITY RPCDBG_MISC

@@ -266,7 +267,7 @@ rpc_proc_init(void)
    dprintk("RPC: registering /proc/net/rpc\n");
    if (!proc_net_rpc) {

```

```

    struct proc_dir_entry *ent;
-   ent = proc_mkdir("rpc", proc_net);
+   ent = proc_mkdir("rpc", per_net(proc_net, init_net()));
    if (ent) {
        ent->owner = THIS_MODULE;
        proc_net_rpc = ent;
@@ -280,7 +281,7 @@ rpc_proc_exit(void)
    dprintk("RPC: unregistering /proc/net/rpc\n");
    if (proc_net_rpc) {
        proc_net_rpc = NULL;
-   remove_proc_entry("net/rpc", NULL);
+   remove_proc_entry("rpc", per_net(proc_net, init_net()));
    }
}

```

```

diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
index 2f208c7..30855e1 100644

```

```

--- a/net/unix/af_unix.c
+++ b/net/unix/af_unix.c
@@ -116,6 +116,7 @@
#include <linux/mount.h>
#include <net/checksum.h>
#include <linux/security.h>
+#include <net/net_namespace.h>

```

```

int sysctl_unix_max_dgram_qlen __read_mostly = 10;

```

```

@@ -2072,7 +2073,7 @@ static int __init af_unix_init(void)

```

```

    sock_register(&unix_family_ops);
#ifdef CONFIG_PROC_FS
-   proc_net_fops_create("unix", 0, &unix_seq_fops);
+   proc_net_fops_create(init_net(), "unix", 0, &unix_seq_fops);
#endif
    unix_sysctl_register();
out:
@@ -2083,7 +2084,7 @@ static void __exit af_unix_exit(void)
{
    sock_unregister(PF_UNIX);
    unix_sysctl_unregister();
-   proc_net_remove("unix");
+   proc_net_remove(init_net(), "unix");
    proto_unregister(&unix_proto);
}

```

```

diff --git a/net/wanrouter/wanproc.c b/net/wanrouter/wanproc.c
index 930ea59..1fcb0b8 100644
--- a/net/wanrouter/wanproc.c

```

```

+++ b/net/wanrouter/wanproc.c
@@ -28,6 +28,7 @@
#include <linux/wanrouter.h> /* WAN router API definitions */
#include <linux/seq_file.h>
#include <linux/smp_lock.h>
+#include <net/net_namespace.h>

#include <asm/io.h>

@@ -287,7 +288,7 @@ static struct file_operations wandev_fops = {
int __init wanrouter_proc_init(void)
{
    struct proc_dir_entry *p;
-   proc_router = proc_mkdir(ROUTER_NAME, proc_net);
+   proc_router = proc_mkdir(ROUTER_NAME, per_net(proc_net, init_net()));
    if (!proc_router)
        goto fail;

@@ -303,7 +304,7 @@ int __init wanrouter_proc_init(void)
fail_stat:
    remove_proc_entry("config", proc_router);
fail_config:
-   remove_proc_entry(ROUTER_NAME, proc_net);
+   remove_proc_entry(ROUTER_NAME, per_net(proc_net, init_net()));
fail:
    return -ENOMEM;
}
@@ -316,7 +317,7 @@ void wanrouter_proc_cleanup(void)
{
    remove_proc_entry("config", proc_router);
    remove_proc_entry("status", proc_router);
-   remove_proc_entry(ROUTER_NAME, proc_net);
+   remove_proc_entry(ROUTER_NAME, per_net(proc_net, init_net()));
}

/*
diff --git a/net/x25/x25_proc.c b/net/x25/x25_proc.c
index a11837d..7bcf98d 100644
--- a/net/x25/x25_proc.c
+++ b/net/x25/x25_proc.c
@@ -20,6 +20,7 @@
#include <linux/init.h>
#include <linux/proc_fs.h>
#include <linux/seq_file.h>
+#include <net/net_namespace.h>
#include <net/sock.h>
#include <net/x25.h>

```

```

@@ -212,7 +213,7 @@ int __init x25_proc_init(void)
    struct proc_dir_entry *p;
    int rc = -ENOMEM;

- x25_proc_dir = proc_mkdir("x25", proc_net);
+ x25_proc_dir = proc_mkdir("x25", per_net(proc_net, init_net()));
    if (!x25_proc_dir)
        goto out;

@@ -231,7 +232,7 @@ out:
    out_socket:
        remove_proc_entry("route", x25_proc_dir);
    out_route:
- remove_proc_entry("x25", proc_net);
+ remove_proc_entry("x25", per_net(proc_net, init_net()));
    goto out;
}

@@ -239,7 +240,7 @@ void __exit x25_proc_exit(void)
{
    remove_proc_entry("route", x25_proc_dir);
    remove_proc_entry("socket", x25_proc_dir);
- remove_proc_entry("x25", proc_net);
+ remove_proc_entry("x25", per_net(proc_net, init_net()));
}

#else /* CONFIG_PROC_FS */
--
1.4.4.1.g278f

```

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