
Subject: [patch 00/20] [Network namespace] Introduction
Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

These patches brings isolation for IPV4 based on a subset of the L2 network namespace.
Multicast and broadcast are not yet implemented.

--

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

Subject: [patch 01/20] [Network namespace] Fix old include header
Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

net/core/fib_rules.c | 3 ++-
1 files changed, 2 insertions(+), 1 deletion(-)

Index: 2.6.19-rc6-mm2/net/core/fib_rules.c

=====

--- 2.6.19-rc6-mm2.orig/net/core/fib_rules.c
+++ 2.6.19-rc6-mm2/net/core/fib_rules.c
@@ -11,7 +11,8 @@
#include <linux/types.h>
#include <linux/kernel.h>
#include <linux/list.h>
-#include <linux/net_ns.h>
+#include <linux/sched.h>
+#include <linux/net_namespace.h>
#include <net/fib_rules.h>

#ifndef CONFIG_NET_NS

--

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

Subject: [patch 02/20] [Network namespace] Delete unused destroy variable

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

net/ipv4/fib_hash.c | 7 +-----

1 files changed, 1 insertion(+), 6 deletions(-)

Index: 2.6.19-rc6-mm2/net/ipv4/fib_hash.c

=====

--- 2.6.19-rc6-mm2.orig/net/ipv4/fib_hash.c

+++ 2.6.19-rc6-mm2/net/ipv4/fib_hash.c

@@ -621,11 +621,6 @@ static int fn_flush_list(struct fn_zone

struct hlist_node *node, *n;

struct fib_node *f;

int found = 0;

-#ifndef CONFIG_NET_NS

- const int destroy = 0;

-#else

- const int destroy = current_net_ns->destroying;

-#endif

hlist_for_each_entry_safe(f, node, n, head, fn_hash) {

struct fib_alias *fa, *fa_node;

@@ -637,7 +632,7 @@ static int fn_flush_list(struct fn_zone

if (fi == NULL)

continue;

- if (destroy || (fi->fib_flags&RTNH_F_DEAD)) {

+ if ((fi->fib_flags&RTNH_F_DEAD)) {

write_lock_bh(&fib_hash_lock);

list_del(&fa->fa_list);

if (list_empty(&f->fn_alias)) {

--

Containers mailing list

Containers@lists.osdl.org

<https://lists.osdl.org/mailman/listinfo/containers>

Subject: [patch 03/20] [Network namespace] Remove useless code

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:20 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

net/core/net_namespace.c | 5 -----
1 files changed, 5 deletions(-)

Index: 2.6.19-rc6-mm2/net/core/net_namespace.c

```
=====
--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c
+++ 2.6.19-rc6-mm2/net/core/net_namespace.c
@@ -128,11 +128,6 @@ void free_net_ns(struct kref *kref)
 /* taking lock after atomic_dec_and_test is racy */
 spin_lock_irqsave(&net_ns_list_lock, flags);
 ns = container_of(kref, struct net_namespace, kref);
- if (atomic_read(&ns->kref.refcount) ||
-     list_empty(&ns->sibling_list)) {
-     spin_unlock_irqrestore(&net_ns_list_lock, flags);
-     return;
- }
 list_del_init(&ns->sibling_list);
 spin_unlock_irqrestore(&net_ns_list_lock, flags);
 put_net_ns(ns->parent);
```

--

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

Subject: [patch 04/20] [Network namespace] Initialize the init network namespace to level 2

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

net/core/net_namespace.c | 1 +
1 files changed, 1 insertion(+)

Index: 2.6.19-rc6-mm2/net/core/net_namespace.c

```
=====
--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c
+++ 2.6.19-rc6-mm2/net/core/net_namespace.c
@@ -26,6 +26,7 @@ struct net_namespace init_net_ns = {
```

```
.pcpu_lstats_p = NULL,  
.child_list = LIST_HEAD_INIT(init_net_ns.child_list),  
.sibling_list = LIST_HEAD_INIT(init_net_ns.sibling_list),  
+ .level      = NET_NS_LEVEL2,  
};
```

```
#ifdef CONFIG_NET_NS
```

```
--
```

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

Subject: [patch 05/20] [Network namespace] Add NS_NET3 to NS_ALL.
Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

include/linux/nsproxy.h | 2 +-
1 files changed, 1 insertion(+), 1 deletion(-)

Index: 2.6.19-rc6-mm2/include/linux/nsproxy.h

```
=====
```

```
--- 2.6.19-rc6-mm2.orig/include/linux/nsproxy.h  
+++ 2.6.19-rc6-mm2/include/linux/nsproxy.h  
@@ -23,7 +23,7 @@ struct user_namespace;  
#define NS_NET2 0x00000010  
#define NS_USER 0x00000020  
#define NS_NET3 0x00000040  
-#define NS_ALL (NS_MNT|NS_UTS|NS_IPC|NS_PID|NS_NET2|NS_USER)  
+#define NS_ALL (NS_MNT|NS_UTS|NS_IPC|NS_PID|NS_NET2|NS_USER|NS_NET3)
```

```
/*  
 * A structure to contain pointers to all per-process
```

```
--
```

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

Subject: [patch 06/20] [Network namespace] Move the nsproxy NULL affection
Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

kernel/nsproxy.c | 2 +-
1 files changed, 1 insertion(+), 1 deletion(-)

Index: 2.6.19-rc6-mm2/kernel/nsproxy.c

=====

```
--- 2.6.19-rc6-mm2.orig/kernel/nsproxy.c
+++ 2.6.19-rc6-mm2/kernel/nsproxy.c
@@ -54,10 +54,10 @@ void exit_task_namespaces(struct task_st
{
    struct nsproxy *ns = p->nsproxy;
    if (ns) {
+   put_nsproxy(ns);
    task_lock(p);
    p->nsproxy = NULL;
    task_unlock(p);
-   put_nsproxy(ns);
    }
}
```

--

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

Subject: [patch 07/20] [Network namespace] Temporary remove the loopback
initialization for layer 3. Allow I3
Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

net/core/net_namespace.c | 32 ++++++-----
1 files changed, 19 insertions(+), 13 deletions(-)

Index: 2.6.19-rc6-mm2/net/core/net_namespace.c

```

=====
--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c
+++ 2.6.19-rc6-mm2/net/core/net_namespace.c
@@ -46,14 +46,12 @@ static struct net_namespace *clone_net_n
    if (current_net_ns->level == NET_NS_LEVEL3)
        return ERR_PTR(-EPERM);

- ns = kmalloc(sizeof(struct net_namespace), GFP_KERNEL);
+ ns = kmemdup(old_ns, sizeof(struct net_namespace), GFP_KERNEL);
    if (!ns)
        return NULL;

    kref_init(&ns->kref);
    ns->ns = old_ns->ns;
- ns->dev_base_p = NULL;
- ns->dev_tail_p = &ns->dev_base_p;
    ns->hash = net_random();
    INIT_LIST_HEAD(&ns->child_list);
    spin_lock_irq(&net_ns_list_lock);
@@ -63,15 +61,20 @@ static struct net_namespace *clone_net_n
    spin_unlock_irq(&net_ns_list_lock);

    if (level == NET_NS_LEVEL2) {
+
+ ns->dev_base_p = NULL;
+ ns->dev_tail_p = &ns->dev_base_p;
+
#ifdef CONFIG_IP_MULTIPLE_TABLES
        INIT_LIST_HEAD(&ns->fib_rules_ops_list);
#endif
        if (ip_fib_struct_init(ns))
            goto out_fib4;
+ if (loopback_init(ns))
+ goto out_loopback;
    }
    ns->level = level;
- if (loopback_init(ns))
- goto out_loopback;
+
    printk(KERN_DEBUG "NET_NS: created new netcontext %p, level %u, "
        "for %s (pid=%d)\n", ns, (ns->level == NET_NS_LEVEL2) ?
        2 : 3, current->comm, current->tgid);
@@ -133,16 +136,19 @@ void free_net_ns(struct kref *kref)
    spin_unlock_irqrestore(&net_ns_list_lock, flags);
    put_net_ns(ns->parent);

- unregister_netdev(ns->loopback_dev_p);
- if (ns->dev_base_p != NULL) {

```

```

- printk("NET_NS: BUG: namespace %p has devices! ref %d\n",
- ns, atomic_read(&ns->kref.refcount));
- return;
- }
- if (ns->level == NET_NS_LEVEL2)
+ if (ns->level == NET_NS_LEVEL2) {
    ip_fib_struct_cleanup();
+ unregister_netdev(ns->loopback_dev_p);
+ if (ns->dev_base_p != NULL) {
+   printk("NET_NS: BUG: namespace %p has devices! ref %d\n",
+         ns, atomic_read(&ns->kref.refcount));
+   return;
+ }
+ }
+
+   printk(KERN_DEBUG "NET_NS: net namespace %p (%u) destroyed\n",
-   ns, ns->id);
+   ns, ns->id);
+
+   kfree(ns);
+ }
+ /* because of put_net_ns() */

```

--

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

Subject: [patch 08/20] [Network namespace] Move the dev name hash relative to the namespace.

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

```

include/linux/net_namespace.h | 14 ++++++++
include/linux/netdevice.h    |  1 +
net/core/dev.c               | 17 ++++++++-----
net/core/net_namespace.c     | 32 ++++++++
4 files changed, 58 insertions(+), 6 deletions(-)

```

Index: 2.6.19-rc6-mm2/net/core/dev.c

```

=====
--- 2.6.19-rc6-mm2.orig/net/core/dev.c

```

```

+++ 2.6.19-rc6-mm2/net/core/dev.c
@@ -188,9 +188,13 @@ EXPORT_SYMBOL(dev_base);
DEFINE_RWLOCK(dev_base_lock);
EXPORT_SYMBOL(dev_base_lock);

-#define NETDEV_HASHBITS 8
+#ifdef CONFIG_NET_NS
+#define dev_name_head (current_net_ns->net_device.name_head)
+#define dev_index_head (current_net_ns->net_device.index_head)
+#else
static struct hlist_head dev_name_head[1<<NETDEV_HASHBITS];
static struct hlist_head dev_index_head[1<<NETDEV_HASHBITS];
+#endif

static inline struct hlist_head *dev_name_hash(const char *name,
struct net_namespace *ns)
@@ -486,13 +490,11 @@ __setup("netdev=", netdev_boot_setup);
struct net_device *__dev_get_by_name(const char *name)
{
struct hlist_node *p;
- struct net_namespace *ns = current_net_ns;
+ struct net_namespace *net_ns = current_net_ns;

- hlist_for_each(p, dev_name_hash(name, ns)) {
+ hlist_for_each(p, dev_name_hash(name, net_ns)) {
struct net_device *dev
= hlist_entry(p, struct net_device, name_hlist);
- if (!net_ns_match(dev->net_ns, ns))
- continue;
if (!strncmp(dev->name, name, IFNAMSIZ))
return dev;
}
@@ -3636,6 +3638,11 @@ static int __init net_dev_init(void)
if (netdev_sysfs_init())
goto out;

+#ifdef CONFIG_NET_NS
+ if (hlist_dev_name_init(current_net_ns))
+ goto out;
+
+ INIT_LIST_HEAD(&ptype_all);
+ for (i = 0; i < 16; i++)
+ INIT_LIST_HEAD(&ptype_base[i]);
Index: 2.6.19-rc6-mm2/net/core/net_namespace.c
=====
--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c
+++ 2.6.19-rc6-mm2/net/core/net_namespace.c

```

```

@@ -12,6 +12,8 @@
#include <linux/net.h>
#include <linux/netdevice.h>
#include <net/ip_fib.h>
+#include <linux/inetdevice.h>
+#include <linux/in.h>

static spinlock_t net_ns_list_lock = SPIN_LOCK_UNLOCKED;

@@ -31,6 +33,33 @@ struct net_namespace init_net_ns = {

#ifdef CONFIG_NET_NS

+int hlist_dev_name_init(struct net_namespace *net_ns)
+{
+ struct hlist_head *hlist_index, *hlist_name;
+ const int size = sizeof(struct hlist_head)*(1<<NETDEV_HASHBITS);
+
+ hlist_name = kmalloc(size, GFP_KERNEL);
+ if (!hlist_name)
+ return -ENOMEM;
+
+ hlist_index = kmalloc(size, GFP_KERNEL);
+ if (!hlist_index) {
+ kfree(hlist_name);
+ return -ENOMEM;
+ }
+
+ net_ns->net_device.name_head = hlist_name;
+ net_ns->net_device.index_head = hlist_index;
+
+ return 0;
+}
+
+static inline void hlist_dev_name_cleanup(struct net_namespace *net_ns)
+{
+ kfree(net_ns->net_device.name_head);
+ kfree(net_ns->net_device.index_head);
+}
+
+/*
+ * Clone a new ns copying an original net ns, setting refcount to 1
+ * @level: level of namespace to create
@@ -52,7 +81,6 @@ static struct net_namespace *clone_net_n

kref_init(&ns->kref);
ns->ns = old_ns->ns;
- ns->hash = net_random();

```

```

INIT_LIST_HEAD(&ns->child_list);
spin_lock_irq(&net_ns_list_lock);
get_net_ns(old_ns);
@@ -62,6 +90,7 @@ static struct net_namespace *clone_net_n

if (level == NET_NS_LEVEL2) {

+ ns->hash = net_random();
  ns->dev_base_p = NULL;
  ns->dev_tail_p = &ns->dev_base_p;

@@ -153,4 +182,5 @@ void free_net_ns(struct kref *kref)
}
/* because of put_net_ns() */
EXPORT_SYMBOL(free_net_ns);
+
#endif /* CONFIG_NET_NS */
Index: 2.6.19-rc6-mm2/include/linux/net_namespace.h
=====
--- 2.6.19-rc6-mm2.orig/include/linux/net_namespace.h
+++ 2.6.19-rc6-mm2/include/linux/net_namespace.h
@@ -5,11 +5,18 @@
#include <linux/nsproxy.h>
#include <linux/errno.h>

+struct net_ns_net_device {
+ struct hlist_head *name_head;
+ struct hlist_head *index_head;
+};
+
struct net_namespace {
struct kref kref;
struct nsproxy *ns;
struct net_device *dev_base_p, **dev_tail_p;
struct net_device *loopback_dev_p;
+ struct net_ns_net_device net_device;
+
struct pcpu_lstats *pcpu_lstats_p;
#ifdef CONFIG_IP_MULTIPLE_TABLES
struct fib_table *fib4_local_table, *fib4_main_table;
@@ -77,6 +84,8 @@ static inline void pop_net_ns(struct net

extern struct net_namespace *find_net_ns(unsigned int id);

+extern int hlist_dev_name_init(struct net_namespace *net_ns);
+
#else /* CONFIG_NET_NS */

```

```
#define INIT_NET_NS(net_ns)
@@ -123,6 +132,11 @@ static inline struct net_namespace *find
    return NULL;
}
```

```
+static inline int net_ns_ioctl(unsigned int cmd, void __user *arg)
+{
+ return 0;
+}
+
#endif /* !CONFIG_NET_NS */
```

```
#endif /* _LINUX_NET_NAMESPACE_H */
```

Index: 2.6.19-rc6-mm2/include/linux/netdevice.h

=====

--- 2.6.19-rc6-mm2.orig/include/linux/netdevice.h

+++ 2.6.19-rc6-mm2/include/linux/netdevice.h

```
@@ -81,6 +81,7 @@ struct netpoll_info;
```

```
#define NETDEV_TX_BUSY 1 /* driver tx path was busy*/
```

```
#define NETDEV_TX_LOCKED -1 /* driver tx lock was already taken */
```

```
+#define NETDEV_HASHBITS 8 /* hash bit size for netdev hash table */
```

```
/*
```

```
 * Compute the worst case header length according to the protocols
```

```
 * used.
```

--

Containers mailing list

Containers@lists.osdl.org

<https://lists.osdl.org/mailman/listinfo/containers>

Subject: [patch 09/20] [Network namespace] Isolate the inet device. ip and ifconfig commands will not show ip

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

include/linux/inetdevice.h | 4 ++++

net/ipv4/devinet.c | 30 ++++++++-----

2 files changed, 28 insertions(+), 6 deletions(-)

Index: 2.6.19-rc6-mm2/include/linux/inetdevice.h

=====

```

--- 2.6.19-rc6-mm2.orig/include/linux/inetdevice.h
+++ 2.6.19-rc6-mm2/include/linux/inetdevice.h
@@ -99,6 +99,7 @@ struct in_ifaddr
    unsigned char ifa_flags;
    unsigned char ifa_prefixlen;
    char ifa_label[IFNAMSIZ];
+ struct net_namespace *ifa_net_ns;
};

extern int register_inetaddr_notifier(struct notifier_block *nb);
@@ -115,6 +116,9 @@ extern __be32 inet_confirm_addr(const s
extern struct in_ifaddr *inet_ifa_byprefix(struct in_device *in_dev, __be32 prefix, __be32 mask);
extern void inet_forward_change(void);

+extern void inet_del_ifa(struct in_device *in_dev, struct in_ifaddr **ifap, int destroy);
+extern void inet_free_ifa(struct in_ifaddr *ifa);
+
static __inline__ int inet_ifa_match(__be32 addr, struct in_ifaddr *ifa)
{
    return !((addr^ifa->ifa_address)&ifa->ifa_mask);
Index: 2.6.19-rc6-mm2/net/ipv4/devinet.c
=====
--- 2.6.19-rc6-mm2.orig/net/ipv4/devinet.c
+++ 2.6.19-rc6-mm2/net/ipv4/devinet.c
@@ -54,6 +54,7 @@
#include <linux/notifier.h>
#include <linux/inetdevice.h>
#include <linux/igmp.h>
+#include <linux/net_namespace.h>
#ifdef CONFIG_SYSCTL
#include <linux/sysctl.h>
#endif
@@ -91,8 +92,6 @@ static struct nla_policy ifa_ipv4_policy
static void rtmsg_ifa(int event, struct in_ifaddr *, struct nlmsg_hdr *, u32);

static BLOCKING_NOTIFIER_HEAD(inetaddr_chain);
-static void inet_del_ifa(struct in_device *in_dev, struct in_ifaddr **ifap,
-    int destroy);
#ifdef CONFIG_SYSCTL
static void devinet_sysctl_register(struct in_device *in_dev,
    struct ipv4_devconf *p);
@@ -120,7 +119,7 @@ static void inet_rcu_free_ifa(struct rcu
    kfree(ifa);
}

-static inline void inet_free_ifa(struct in_ifaddr *ifa)
+void inet_free_ifa(struct in_ifaddr *ifa)
{

```

```

    call_rcu(&ifa->rcu_head, inet_rcu_free_ifa);
}
@@ -268,6 +267,7 @@ static void __inet_del_ifa(struct in_dev

    if (!(ifa->ifa_flags & IFA_F_SECONDARY) ||
        ifa1->ifa_mask != ifa->ifa_mask ||
+    !net_ns_match(ifa->ifa_net_ns, ifa1->ifa_net_ns) ||
        !inet_ifa_match(ifa1->ifa_address, ifa)) {
        ifap1 = &ifa->ifa_next;
        prev_prom = ifa;
@@ -333,8 +333,8 @@ static void __inet_del_ifa(struct in_dev
    }
}

-static void inet_del_ifa(struct in_device *in_dev, struct in_ifaddr **ifap,
-    int destroy)
+void inet_del_ifa(struct in_device *in_dev, struct in_ifaddr **ifap,
+    int destroy)
{
    __inet_del_ifa(in_dev, ifap, destroy, NULL, 0);
}
@@ -470,6 +470,9 @@ static int inet_rtm_deladdr(struct sk_bu

    for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
         ifap = &ifa->ifa_next) {
+    if (!net_ns_match(ifa->ifa_net_ns, current_net_ns))
+        continue;
+
        if (tb[IFA_LOCAL] &&
            ifa->ifa_local != nla_get_be32(tb[IFA_LOCAL]))
            continue;
@@ -543,6 +546,7 @@ static struct in_ifaddr *rtm_to_ifaddr(s
    ifa->ifa_flags = ifm->ifa_flags;
    ifa->ifa_scope = ifm->ifa_scope;
    ifa->ifa_dev = in_dev;
+ ifa->ifa_net_ns = current_net_ns;

    ifa->ifa_local = nla_get_be32(tb[IFA_LOCAL]);
    ifa->ifa_address = nla_get_be32(tb[IFA_ADDRESS]);
@@ -688,6 +692,8 @@ int devinet_ioctl(unsigned int cmd, void
    for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
         ifap = &ifa->ifa_next) {
        if (!strcmp(ifr.ifr_name, ifa->ifa_label) &&
+    net_ns_match(ifa->ifa_net_ns,
+    current_net_ns) &&
        sin_orig.sin_addr.s_addr ==
            ifa->ifa_address) {
            break; /* found */

```

```

@@ -700,11 +706,16 @@ int devinet_ioctl(unsigned int cmd, void
    if (!ifa) {
        for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
            ifap = &ifa->ifa_next)
-       if (!strcmp(ifr.ifr_name, ifa->ifa_label))
+       if (!strcmp(ifr.ifr_name, ifa->ifa_label) &&
+           net_ns_match(ifa->ifa_net_ns,
+               current_net_ns))
            break;
    }
}

+ if (ifa && !net_ns_match(ifa->ifa_net_ns, current_net_ns))
+ goto done;
+
ret = -EADDRNOTAVAIL;
if (ifa && cmd != SIOCSIFADDR && cmd != SIOCSIFFLAGS)
goto done;
@@ -748,6 +759,8 @@ int devinet_ioctl(unsigned int cmd, void
ret = -ENOBUFS;
if ((ifa = inet_alloc_ifa()) == NULL)
break;
+
+ ifa->ifa_net_ns = current_net_ns;
+ if (colon)
+     memcpy(ifa->ifa_label, ifr.ifr_name, IFNAMSIZ);
+ else
@@ -852,6 +865,8 @@ static int inet_gifconf(struct net_devic
goto out;

for (; ifa; ifa = ifa->ifa_next) {
+ if (!net_ns_match(ifa->ifa_net_ns, current_net_ns))
+ continue;
    if (!buf) {
        done += sizeof(ifr);
        continue;
@@ -1085,6 +1100,7 @@ static int inetdev_event(struct notifier
in_dev_hold(in_dev);
ifa->ifa_dev = in_dev;
ifa->ifa_scope = RT_SCOPE_HOST;
+ ifa->ifa_net_ns = current_net_ns;
+ memcpy(ifa->ifa_label, dev->name, IFNAMSIZ);
+ inet_insert_ifa(ifa);
    }
}
@@ -1197,6 +1213,8 @@ static int inet_dump_ifaddr(struct sk_bu

for (ifa = in_dev->ifa_list, ip_idx = 0; ifa;
    ifa = ifa->ifa_next, ip_idx++) {

```

```

+ if (!net_ns_match(ifa->ifa_net_ns, current_net_ns))
+ continue;
+ if (ip_idx < s_ip_idx)
+ continue;
+ if (inet_fill_ifaddr(skb, ifa, NETLINK_CB(cb->skb).pid,
--

```

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

Subject: [patch 10/20] [Network namespace] ioctl to push ifa to net_ns l3
Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

New ioctl to "push" ifaddr to a container. Actually, the push is done from the current namespace, so the right word is "pull". That will be changed to move ifaddr from l2 network namespace to l3.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

```

include/linux/net_namespace.h | 4 +
include/linux/sockios.h      | 4 +
net/core/net_namespace.c     | 97 +++++
net/ipv4/af_inet.c           | 4 +
4 files changed, 108 insertions(+), 1 deletion(-)

```

Index: 2.6.19-rc6-mm2/include/linux/sockios.h

=====

--- 2.6.19-rc6-mm2.orig/include/linux/sockios.h

+++ 2.6.19-rc6-mm2/include/linux/sockios.h

@@ -122,6 +122,10 @@

#define SIOCBRADDIF 0x89a2 /* add interface to bridge */

#define SIOCBRDELIF 0x89a3 /* remove interface from bridge */

/* Container calls */

+#define SIOCNETNSPUSHIF 0x89b0 /* add ifaddr to namespace */

+#define SIOCNETNSPULLIF 0x89b1 /* remove ifaddr to namespace */

+

/* Device private ioctl calls */

/*

Index: 2.6.19-rc6-mm2/net/ipv4/af_inet.c

=====

```

--- 2.6.19-rc6-mm2.orig/net/ipv4/af_inet.c
+++ 2.6.19-rc6-mm2/net/ipv4/af_inet.c
@@ -789,6 +789,10 @@ int inet_ioctl(struct socket *sock, unsi
    case SIOCSIFFLAGS:
        err = devinet_ioctl(cmd, (void __user *)arg);
        break;
+    case SIOCNETNSPUSHIF:
+    case SIOCNETNSPULLIF:
+    err = net_ns_ioctl(cmd, (void __user *)arg);
+    break;
    default:
        if (sk->sk_prot->ioctl)
            err = sk->sk_prot->ioctl(sk, cmd, arg);
Index: 2.6.19-rc6-mm2/include/linux/net_namespace.h
=====
--- 2.6.19-rc6-mm2.orig/include/linux/net_namespace.h
+++ 2.6.19-rc6-mm2/include/linux/net_namespace.h
@@ -86,6 +86,8 @@ extern struct net_namespace *find_net_ns

extern int hlist_dev_name_init(struct net_namespace *net_ns);

+extern int net_ns_ioctl(unsigned int cmd, void __user *arg);
+
#else /* CONFIG_NET_NS */

#define INIT_NET_NS(net_ns)
@@ -134,7 +136,7 @@ static inline struct net_namespace *find

static inline int net_ns_ioctl(unsigned int cmd, void __user *arg)
{
- return 0;
+ return -ENOSYS;
}

#endif /* !CONFIG_NET_NS */
Index: 2.6.19-rc6-mm2/net/core/net_namespace.c
=====
--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c
+++ 2.6.19-rc6-mm2/net/core/net_namespace.c
@@ -153,6 +153,28 @@ int copy_net_ns(int flags, struct task_s
    return err;
}

+static void release_ifa_to_parent(const struct net_namespace* net_ns)
+{
+ struct net_device *dev;
+ struct in_device *in_dev;
+

```

```

+ read_lock(&dev_base_lock);
+ rcu_read_lock();
+ for (dev = dev_base; dev; dev = dev->next) {
+   in_dev = __in_dev_get_rcu(dev);
+   if (!in_dev)
+     continue;
+
+   for_ifa(in_dev) {
+     if (ifa->ifa_net_ns != net_ns)
+       continue;
+     ifa->ifa_net_ns = net_ns->parent;
+   } endfor_ifa(in_dev);
+ }
+ read_unlock(&dev_base_lock);
+ rcu_read_unlock();
+}
+
void free_net_ns(struct kref *kref)
{
    struct net_namespace *ns;
@@ -175,6 +197,10 @@ void free_net_ns(struct kref *kref)
}
}

+ if (ns->level == NET_NS_LEVEL3) {
+   release_ifa_to_parent(ns);
+ }
+
    printk(KERN_DEBUG "NET_NS: net namespace %p (%u) destroyed\n",
           ns, ns->id);

@@ -183,4 +209,75 @@ void free_net_ns(struct kref *kref)
/* because of put_net_ns() */
EXPORT_SYMBOL(free_net_ns);

+int net_ns_ioctl(unsigned int cmd, void __user *arg)
+{
+   struct ifreq ifr;
+   struct sockaddr_in *sin = (struct sockaddr_in *)&ifr.ifr_addr;
+   struct net_namespace *net_ns = current_net_ns;
+   struct net_device *dev;
+   struct in_device *in_dev;
+   struct in_ifaddr **ifap = NULL;
+   struct in_ifaddr *ifa = NULL;
+   char *colon;
+   int err;
+
+   if (!capable(CAP_NET_ADMIN))

```

```

+ return -EPERM;
+
+ if (net_ns->level != NET_NS_LEVEL3)
+ return -EPERM;
+
+ if (copy_from_user(&ifr, arg, sizeof(struct ifreq)))
+ return -EFAULT;
+
+ ifr.ifr_name[IFNAMSIZ - 1] = 0;
+
+
+ colon = strchr(ifr.ifr_name, ':');
+ if (colon)
+ *colon = 0;
+
+ rtnl_lock();
+
+ err = -ENODEV;
+ dev = __dev_get_by_name(ifr.ifr_name);
+ if (!dev)
+ goto out;
+
+ if (colon)
+ *colon = ':';
+
+ err = -EADDRNOTAVAIL;
+ in_dev = __in_dev_get_rtnl(dev);
+ if (!in_dev)
+ goto out;
+
+ for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
+      ifap = &ifa->ifa_next)
+ if (!strcmp(ifr.ifr_name, ifa->ifa_label) &&
+     sin->sin_addr.s_addr == ifa->ifa_local)
+ break;
+ if (!ifa)
+ goto out;
+
+ err = -EINVAL;
+ switch(cmd) {
+
+ case SIOCNETHNSPUSHIF:
+ ifa->ifa_net_ns = net_ns;
+ break;
+
+ case SIOCNETHNSPULLIF:
+ ifa->ifa_net_ns = net_ns->parent;
+ break;

```

```

+ default:
+ goto out;
+ }
+
+ err = 0;
+out:
+ rtnl_unlock();
+ return err;
+}
+
#endif /* CONFIG_NET_NS */

```

--

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

Subject: [patch 11/20] [Network namespace] Check the bind address is allowed. It must match ifaddr assigned t

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

```

include/linux/net_namespace.h | 7 +++++++
net/core/net_namespace.c      | 40 +++++++++++++++++++++++++++++++++++++
net/ipv4/af_inet.c            | 2 ++
net/ipv4/raw.c                | 3 +++
4 files changed, 52 insertions(+)

```

Index: 2.6.19-rc6-mm2/net/ipv4/af_inet.c

=====

```

--- 2.6.19-rc6-mm2.orig/net/ipv4/af_inet.c
+++ 2.6.19-rc6-mm2/net/ipv4/af_inet.c
@@ -433,6 +433,8 @@ int inet_bind(struct socket *sock, struc
 * is temporarily down)
 */
err = -EADDRNOTAVAIL;
+ if (net_ns_check_bind(chk_addr_ret, addr->sin_addr.s_addr))
+ goto out;
if (!sysctl_ip_nonlocal_bind &&
    !inet->freebind &&
    addr->sin_addr.s_addr != INADDR_ANY &&

```

Index: 2.6.19-rc6-mm2/net/ipv4/raw.c

```

=====
--- 2.6.19-rc6-mm2.orig/net/ipv4/raw.c
+++ 2.6.19-rc6-mm2/net/ipv4/raw.c
@@ -559,7 +559,10 @@ static int raw_bind(struct sock *sk, str
    if (sk->sk_state != TCP_CLOSE || addr_len < sizeof(struct sockaddr_in))
        goto out;
    chk_addr_ret = inet_addr_type(addr->sin_addr.s_addr);
+
    ret = -EADDRNOTAVAIL;
+ if (net_ns_check_bind(chk_addr_ret, addr->sin_addr.s_addr))
+ goto out;
    if (addr->sin_addr.s_addr && chk_addr_ret != RTN_LOCAL &&
        chk_addr_ret != RTN_MULTICAST && chk_addr_ret != RTN_BROADCAST)
        goto out;
Index: 2.6.19-rc6-mm2/include/linux/net_namespace.h
=====
--- 2.6.19-rc6-mm2.orig/include/linux/net_namespace.h
+++ 2.6.19-rc6-mm2/include/linux/net_namespace.h
@@ -88,6 +88,8 @@ extern int hlist_dev_name_init(struct ne

extern int net_ns_ioctl(unsigned int cmd, void __user *arg);

+extern int net_ns_check_bind(int addr_type, u32 addr);
+
#else /* CONFIG_NET_NS */

#define INIT_NET_NS(net_ns)
@@ -139,6 +141,11 @@ static inline int net_ns_ioctl(unsigned
    return -ENOSYS;
}

+static inline int net_ns_check_bind(int addr_type, u32 addr)
+{
+ return 0;
+}
+
#endif /* !CONFIG_NET_NS */

#endif /* _LINUX_NET_NAMESPACE_H */
Index: 2.6.19-rc6-mm2/net/core/net_namespace.c
=====
--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c
+++ 2.6.19-rc6-mm2/net/core/net_namespace.c
@@ -280,4 +280,44 @@ out:
    return err;
}

+int net_ns_check_bind(int addr_type, u32 addr)

```

```

+{
+ int ret = -EPERM;
+     struct net_device *dev;
+     struct in_device *in_dev;
+ struct net_namespace *net_ns = current_net_ns;
+
+ if (LOOPBACK(addr) ||
+     MULTICAST(addr) ||
+     INADDR_ANY == addr ||
+     INADDR_BROADCAST == addr)
+ return 0;
+
+     read_lock(&dev_base_lock);
+     rcu_read_lock();
+     for (dev = dev_base; dev; dev = dev->next) {
+         in_dev = __in_dev_get_rcu(dev);
+         if (!in_dev)
+             continue;
+
+         for_ifa(in_dev) {
+             if (ifa->ifa_net_ns != net_ns)
+ continue;
+         if (addr == ifa->ifa_local ||
+             addr == ifa->ifa_broadcast ||
+             addr == (ifa->ifa_local & ifa->ifa_mask) ||
+             addr == ((ifa->ifa_address & ifa->ifa_mask) |
+                 ~ifa->ifa_mask)) {
+             ret = 0;
+             goto out;
+         }
+     } endfor_ifa(in_dev);
+ }
+out:
+     read_unlock(&dev_base_lock);
+     rcu_read_unlock();
+
+ return ret;
+}
+
+endif /* CONFIG_NET_NS */

```

--

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

Subject: [patch 12/20] [Network namespace] When no source address is specified, search from the dev list the

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

```
include/linux/net_namespace.h | 14 ++++++++
net/core/net_namespace.c      | 55 +++++++++++++++++++++++++++++++++++++
net/ipv4/route.c              | 27 ++++++++-----
3 files changed, 86 insertions(+), 10 deletions(-)
```

Index: 2.6.19-rc6-mm2/net/ipv4/route.c

=====

--- 2.6.19-rc6-mm2.orig/net/ipv4/route.c

+++ 2.6.19-rc6-mm2/net/ipv4/route.c

@@ -2472,17 +2472,17 @@ static int ip_route_output_slow(struct r

```
    if (LOCAL_MCAST(oldflp->fl4_dst) || oldflp->fl4_dst == htonl(0xFFFFFFFF)) {
        if (!fl.fl4_src)
-       fl.fl4_src = inet_select_addr(dev_out, 0,
-       RT_SCOPE_LINK);
+       fl.fl4_src = SELECT_SRC_ADDR(dev_out, 0,
+       RT_SCOPE_LINK);
        goto make_route;
    }
    if (!fl.fl4_src) {
        if (MULTICAST(oldflp->fl4_dst))
-       fl.fl4_src = inet_select_addr(dev_out, 0,
-       fl.fl4_scope);
+       fl.fl4_src = SELECT_SRC_ADDR(dev_out, 0,
+       fl.fl4_scope);
        else if (!oldflp->fl4_dst)
-       fl.fl4_src = inet_select_addr(dev_out, 0,
-       RT_SCOPE_HOST);
+       fl.fl4_src = SELECT_SRC_ADDR(dev_out, 0,
+       RT_SCOPE_HOST);
    }
}
```

@@ -2522,8 +2522,8 @@ static int ip_route_output_slow(struct r

*/

```
    if (fl.fl4_src == 0)
-       fl.fl4_src = inet_select_addr(dev_out, 0,
-       RT_SCOPE_LINK);
+       fl.fl4_src = SELECT_SRC_ADDR(dev_out, 0,
```

```

+      RT_SCOPE_LINK);
  res.type = RTN_UNICAST;
  goto make_route;
}
@@ -2536,7 +2536,12 @@ static int ip_route_output_slow(struct r

```

```

  if (res.type == RTN_LOCAL) {
    if (!fl.fl4_src)
+ifdef CONFIG_NET
+  fl.fl4_src = net_ns_select_source_address(dev_out, 0,
+      RT_SCOPE_LINK);
+else
    fl.fl4_src = fl.fl4_dst;
+endif
    if (dev_out)
      dev_put(dev_out);
    dev_out = &loopback_dev;
@@ -2558,8 +2563,10 @@ static int ip_route_output_slow(struct r
  fib_select_default(&fl, &res);

```

```

  if (!fl.fl4_src)
- fl.fl4_src = FIB_RES_PREFSRC(res);
-
+ fl.fl4_src = res.fi->fib_prefsrc ? :
+ SELECT_SRC_ADDR(FIB_RES_DEV(res),
+ FIB_RES_GW(res),
+ res.scope);
  if (dev_out)
    dev_put(dev_out);
  dev_out = FIB_RES_DEV(res);

```

Index: 2.6.19-rc6-mm2/include/linux/net_namespace.h

=====

--- 2.6.19-rc6-mm2.orig/include/linux/net_namespace.h

+++ 2.6.19-rc6-mm2/include/linux/net_namespace.h

@@ -4,6 +4,7 @@

#include <linux/kref.h>

#include <linux/nsproxy.h>

#include <linux/errno.h>

+#include <linux/types.h>

```

struct net_ns_net_device {
    struct hlist_head *name_head;

```

@@ -90,6 +91,11 @@ extern int net_ns_ioctl(unsigned int cmd

extern int net_ns_check_bind(int addr_type, u32 addr);

```

+extern __be32 net_ns_select_source_address(const struct net_device *dev,
+    u32 dst, int scope);

```

```

+
+#define SELECT_SRC_ADDR net_ns_select_source_address
+
+else /* CONFIG_NET_NS */

#define INIT_NET_NS(net_ns)
@@ -146,6 +152,14 @@ static inline int net_ns_check_bind(int
    return 0;
}

+static inline __be32 net_ns_select_source_address(struct net_device *dev,
+    u32 dst, int scope)
+{
+    return 0;
+}
+
+#define SELECT_SRC_ADDR inet_select_addr
+
+endif /* !CONFIG_NET_NS */

+endif /* _LINUX_NET_NAMESPACE_H */
Index: 2.6.19-rc6-mm2/net/core/net_namespace.c
=====
--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c
+++ 2.6.19-rc6-mm2/net/core/net_namespace.c
@@ -320,4 +320,59 @@ out:
    return ret;
}

+__be32 net_ns_select_source_address(const struct net_device *dev,
+    u32 dst, int scope)
+{
+    __be32 addr = 0;
+    struct in_device *in_dev;
+    struct net_namespace *net_ns = current_net_ns;
+
+    if (!dev)
+        goto no_dev;
+
+    rcu_read_lock();
+    in_dev = __in_dev_get_rcu(dev);
+    if (!in_dev)
+        goto no_in_dev;
+
+    for_ifa(in_dev) {
+        if (ifa->ifa_scope > scope)
+            continue;
+        if (ifa->ifa_net_ns != net_ns)

```

```

+ continue;
+ if (!dst || inet_ifa_match(dst, ifa)) {
+   addr = ifa->ifa_local;
+   break;
+ }
+ if (!addr)
+   addr = ifa->ifa_local;
+ } endfor_ifa(in_dev);
+no_in_dev:
+ rcu_read_unlock();
+
+ if (addr)
+   goto out;
+
+no_dev:
+ read_lock(&dev_base_lock);
+ rcu_read_lock();
+ for (dev = dev_base; dev; dev = dev->next) {
+   if ((in_dev = __in_dev_get_rcu(dev)) == NULL)
+     continue;
+
+   for_ifa(in_dev) {
+     if (ifa->ifa_scope != RT_SCOPE_LINK &&
+         ifa->ifa_scope <= scope &&
+         ifa->ifa_net_ns == net_ns) {
+       addr = ifa->ifa_local;
+       goto out_unlock_both;
+     }
+   } endfor_ifa(in_dev);
+ }
+out_unlock_both:
+ read_unlock(&dev_base_lock);
+ rcu_read_unlock();
+out:
+ return addr;
+}
#endif /* CONFIG_NET_NS */

--

```

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

Subject: [patch 13/20] [Network namespace] fix silly ifdef error
Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:30 GMT

Replace-Subject: fix silly ifdef error

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

net/ipv4/route.c | 2 +-
1 files changed, 1 insertion(+), 1 deletion(-)

Index: 2.6.19-rc6-mm2/net/ipv4/route.c

=====

--- 2.6.19-rc6-mm2.orig/net/ipv4/route.c
+++ 2.6.19-rc6-mm2/net/ipv4/route.c
@@ -2536,7 +2536,7 @@ static int ip_route_output_slow(struct r

```
    if (res.type == RTN_LOCAL) {  
        if (!fl.fl4_src)  
-#ifdef CONFIG_NET  
+#ifdef CONFIG_NET_NS  
            fl.fl4_src = net_ns_select_source_address(dev_out, 0,  
                RT_SCOPE_LINK);  
    #else
```

--

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

Subject: [patch 14/20] [Network namespace] Switch the l3 namespace using the destination address. Does not wo

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

include/linux/net_namespace.h | 7 +++++++
net/core/net_namespace.c | 28 ++++++
net/ipv4/ip_input.c | 21 ++++++
3 files changed, 55 insertions(+), 1 deletion(-)

Index: 2.6.19-rc6-mm2/net/ipv4/ip_input.c

=====

```

--- 2.6.19-rc6-mm2.orig/net/ipv4/ip_input.c
+++ 2.6.19-rc6-mm2/net/ipv4/ip_input.c
@@ -374,6 +374,11 @@ int ip_rcv(struct sk_buff *skb, struct n
{
    struct iphdr *iph;
    u32 len;
+ int err;
+ #ifdef CONFIG_NET_NS
+ struct net_namespace *net_ns = current_net_ns;
+ struct net_namespace *dst_net_ns = NULL;
+ #endif

    /* When the interface is in promisc. mode, drop all the crap
     * that it receives, do not try to analyse it.
@@ -393,6 +398,11 @@ int ip_rcv(struct sk_buff *skb, struct n

    iph = skb->nh.iph;

+ #ifdef CONFIG_NET_NS
+ dst_net_ns = net_ns_find_from_dest_addr(iph->daddr);
+ if (dst_net_ns && net_ns != dst_net_ns)
+ push_net_ns(dst_net_ns, net_ns);
+ #endif
    /*
     * RFC1122: 3.1.2.2 MUST silently discard any IP frame that fails the checksum.
     */
@@ -431,10 +441,19 @@ int ip_rcv(struct sk_buff *skb, struct n
    /* Remove any debris in the socket control block */
    memset(IPCB(skb), 0, sizeof(struct inet_skb_parm));

- return NF_HOOK(PF_INET, NF_IP_PRE_ROUTING, skb, dev, NULL,
+ err = NF_HOOK(PF_INET, NF_IP_PRE_ROUTING, skb, dev, NULL,
    ip_rcv_finish);
+ #ifdef CONFIG_NET_NS
+ if (dst_net_ns && net_ns != dst_net_ns)
+ pop_net_ns(net_ns);
+ #endif
+ return err;

    inhdr_error:
+ #ifdef CONFIG_NET_NS
+ if (dst_net_ns && net_ns != dst_net_ns)
+ pop_net_ns(net_ns);
+ #endif
    IP_INC_STATS_BH(IPSTATS_MIB_INHDRERRORS);
    drop:
    kfree_skb(skb);
Index: 2.6.19-rc6-mm2/include/linux/net_namespace.h

```

```

=====
--- 2.6.19-rc6-mm2.orig/include/linux/net_namespace.h
+++ 2.6.19-rc6-mm2/include/linux/net_namespace.h
@@ -94,6 +94,8 @@ extern int net_ns_check_bind(int addr_ty
extern __be32 net_ns_select_source_address(const struct net_device *dev,
u32 dst, int scope);

+extern struct net_namespace *net_ns_find_from_dest_addr(u32 daddr);
+
#define SELECT_SRC_ADDR net_ns_select_source_address

#else /* CONFIG_NET_NS */
@@ -158,6 +160,11 @@ static inline __be32 net_ns_select_sourc
return 0;
}

+static inline struct net_namespace *net_ns_find_from_dest_addr(u32 daddr)
+{
+ return current_net_ns;
+}
+
#define SELECT_SRC_ADDR inet_select_addr

#endif /* !CONFIG_NET_NS */
Index: 2.6.19-rc6-mm2/net/core/net_namespace.c
=====
--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c
+++ 2.6.19-rc6-mm2/net/core/net_namespace.c
@@ -375,4 +375,32 @@ out:
return addr;
}
+
+struct net_namespace *net_ns_find_from_dest_addr(u32 daddr)
+{
+ struct net_namespace *net_ns = NULL;
+ struct net_device *dev;
+ struct in_device *in_dev;
+
+ if (LOOPBACK(daddr))
+ return current_net_ns;
+
+ read_lock(&dev_base_lock);
+ rcu_read_lock();
+ for (dev = dev_base; dev; dev = dev->next) {
+ if ((in_dev = __in_dev_get_rcu(dev)) == NULL)
+ continue;
+ for_ifa(in_dev) {

```

```

+ if (ifa->ifa_local == daddr) {
+   net_ns = ifa->ifa_net_ns;
+   goto out_unlock_both;
+ }
+ } endfor_ifa(in_dev);
+ }
+out_unlock_both:
+ read_unlock(&dev_base_lock);
+ rcu_read_unlock();
+
+ return net_ns;
+}
#endif /* CONFIG_NET_NS */

```

--

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

Subject: [patch 15/20] [Network namespace] Add visibility on the loopback address.
Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

```

include/linux/net_namespace.h | 9 ++++++++
net/core/net_namespace.c      | 14 ++++++++
net/ipv4/devinet.c            | 12 +++++-----
3 files changed, 28 insertions(+), 7 deletions(-)

```

Index: 2.6.19-rc6-mm2/net/ipv4/devinet.c

=====

```

--- 2.6.19-rc6-mm2.orig/net/ipv4/devinet.c
+++ 2.6.19-rc6-mm2/net/ipv4/devinet.c
@@ -692,8 +692,7 @@ int devinet_ioctl(unsigned int cmd, void
     for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
         ifap = &ifa->ifa_next) {
     if (!strcmp(ifr.ifr_name, ifa->ifa_label) &&
-        net_ns_match(ifa->ifa_net_ns,
-        current_net_ns) &&
+        net_ns_ifa_is_visible(ifa) &&
         sin_orig.sin_addr.s_addr ==
         ifa->ifa_address) {
         break; /* found */

```

```

@@ -707,13 +706,12 @@ int devinet_ioctl(unsigned int cmd, void
    for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
        ifap = &ifa->ifa_next)
        if (!strcmp(ifr.ifr_name, ifa->ifa_label) &&
-           net_ns_match(ifa->ifa_net_ns,
-           current_net_ns))
+           net_ns_ifa_is_visible(ifa))
            break;
    }
}

```

```

- if (ifa && !net_ns_match(ifa->ifa_net_ns, current_net_ns))
+ if (ifa && !net_ns_ifa_is_visible(ifa))
    goto done;

```

```

ret = -EADDRNOTAVAIL;
@@ -865,7 +863,7 @@ static int inet_gifconf(struct net_devic
    goto out;

```

```

    for (; ifa; ifa = ifa->ifa_next) {
-   if (!net_ns_match(ifa->ifa_net_ns, current_net_ns))
+   if (!net_ns_ifa_is_visible(ifa))
        continue;
        if (!buf) {
            done += sizeof(ifr);
@@ -1213,7 +1211,7 @@ static int inet_dump_ifaddr(struct sk_bu

```

```

    for (ifa = in_dev->ifa_list, ip_idx = 0; ifa;
        ifa = ifa->ifa_next, ip_idx++) {
-   if (!net_ns_match(ifa->ifa_net_ns, current_net_ns))
+   if (!net_ns_ifa_is_visible(ifa))
        continue;
        if (ip_idx < s_ip_idx)
            continue;

```

Index: 2.6.19-rc6-mm2/include/linux/net_namespace.h

=====

--- 2.6.19-rc6-mm2.orig/include/linux/net_namespace.h

+++ 2.6.19-rc6-mm2/include/linux/net_namespace.h

@@ -6,6 +6,8 @@

#include <linux/errno.h>

#include <linux/types.h>

+struct in_ifaddr;

+

```

struct net_ns_net_device {
    struct hlist_head *name_head;
    struct hlist_head *index_head;

```

@@ -96,6 +98,8 @@ extern __be32 net_ns_select_source_addre

```
extern struct net_namespace *net_ns_find_from_dest_addr(u32 daddr);
```

```
+extern int net_ns_ifa_is_visible(const struct in_ifaddr *ifa);
```

```
+
```

```
#define SELECT_SRC_ADDR net_ns_select_source_address
```

```
#else /* CONFIG_NET_NS */
```

```
@@ -165,6 +169,11 @@ static inline struct net_namespace *net_
```

```
return current_net_ns;
```

```
}
```

```
+static inline int net_ns_ifa_is_visible(const struct in_ifaddr *ifa)
```

```
+{
```

```
+ return 1;
```

```
+}
```

```
+
```

```
#define SELECT_SRC_ADDR inet_select_addr
```

```
#endif /* !CONFIG_NET_NS */
```

```
Index: 2.6.19-rc6-mm2/net/core/net_namespace.c
```

```
=====
```

```
--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c
```

```
+++ 2.6.19-rc6-mm2/net/core/net_namespace.c
```

```
@@ -403,4 +403,18 @@ out_unlock_both:
```

```
return net_ns;
```

```
}
```

```
+
```

```
+int net_ns_ifa_is_visible(const struct in_ifaddr *ifa)
```

```
+{
```

```
+ struct net_namespace *net_ns = current_net_ns;
```

```
+
```

```
+ if (LOOPBACK(ifa->ifa_local))
```

```
+ return 1;
```

```
+
```

```
+ if (net_ns_match(ifa->ifa_net_ns, net_ns))
```

```
+ return 1;
```

```
+
```

```
+ return 0;
```

```
+}
```

```
+
```

```
#endif /* CONFIG_NET_NS */
```

```
--
```

Containers mailing list

Containers@lists.osdl.org

Subject: [patch 16/20] [Network namespace] Add loopback isolation.

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

```
include/linux/net_namespace.h | 7 ++++++--
include/linux/skbuff.h        | 5 ++++-
net/core/net_namespace.c      | 13 ++++++++----
net/ipv4/ip_input.c           | 4 +++-
net/ipv4/ip_output.c          | 2 ++
5 files changed, 23 insertions(+), 8 deletions(-)
```

Index: 2.6.19-rc6-mm2/include/linux/skbuff.h

=====

--- 2.6.19-rc6-mm2.orig/include/linux/skbuff.h

+++ 2.6.19-rc6-mm2/include/linux/skbuff.h

@@ -227,6 +227,7 @@ enum {

* @dma_cookie: a cookie to one of several possible DMA operations

* done by skb DMA functions

* @secmark: security marking

+ * @net_ns: namespace destination

*/

struct sk_buff {

@@ -311,7 +312,9 @@ struct sk_buff {

#ifdef CONFIG_NETWORK_SECMARK

__u32 secmark;

#endif

-

+ #ifdef CONFIG_NET_NS

+ struct net_namespace *net_ns;

+ #endif

__u32 mark;

/* These elements must be at the end, see alloc_skb() for details. */

Index: 2.6.19-rc6-mm2/net/ipv4/ip_input.c

=====

--- 2.6.19-rc6-mm2.orig/net/ipv4/ip_input.c

+++ 2.6.19-rc6-mm2/net/ipv4/ip_input.c

@@ -399,9 +399,9 @@ int ip_rcv(struct sk_buff *skb, struct n

iph = skb->nh.iph;

```

#ifdef CONFIG_NET_NS
- dst_net_ns = net_ns_find_from_dest_addr(iph->daddr);
+ dst_net_ns = net_ns_find_from_dest_addr(skb);
  if (dst_net_ns && net_ns != dst_net_ns)
- push_net_ns(dst_net_ns, net_ns);
+ net_ns = push_net_ns(dst_net_ns);
#endif
/*
  * RFC1122: 3.1.2.2 MUST silently discard any IP frame that fails the checksum.

```

Index: 2.6.19-rc6-mm2/net/ipv4/ip_output.c

```

=====
--- 2.6.19-rc6-mm2.orig/net/ipv4/ip_output.c
+++ 2.6.19-rc6-mm2/net/ipv4/ip_output.c
@@ -277,9 +277,11 @@ int ip_mc_output(struct sk_buff *skb)
int ip_output(struct sk_buff *skb)
{
  struct net_device *dev = skb->dst->dev;
+ struct net_namespace *net_ns = current_net_ns;

  IP_INC_STATS(IPSTATS_MIB_OUTREQUESTS);

+ skb->net_ns = net_ns;
  skb->dev = dev;
  skb->protocol = htons(ETH_P_IP);

```

Index: 2.6.19-rc6-mm2/include/linux/net_namespace.h

```

=====
--- 2.6.19-rc6-mm2.orig/include/linux/net_namespace.h
+++ 2.6.19-rc6-mm2/include/linux/net_namespace.h
@@ -7,6 +7,7 @@
#include <linux/types.h>

struct in_ifaddr;
+struct sk_buff;

struct net_ns_net_device {
  struct hlist_head *name_head;
@@ -96,7 +97,8 @@ extern int net_ns_check_bind(int addr_ty
extern __be32 net_ns_select_source_address(const struct net_device *dev,
  u32 dst, int scope);

-extern struct net_namespace *net_ns_find_from_dest_addr(u32 daddr);
+extern struct net_namespace
+*net_ns_find_from_dest_addr(const struct sk_buff *skb);

extern int net_ns_ifa_is_visible(const struct in_ifaddr *ifa);

@@ -164,7 +166,8 @@ static inline __be32 net_ns_select_sourc

```

```

    return 0;
}

```

```

-static inline struct net_namespace *net_ns_find_from_dest_addr(u32 daddr)
+static inline struct net_namespace
+*net_ns_find_from_dest_addr(const struct sk_buff *skb)
{
    return current_net_ns;
}

```

Index: 2.6.19-rc6-mm2/net/core/net_namespace.c

=====

--- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c

+++ 2.6.19-rc6-mm2/net/core/net_namespace.c

@@ -14,6 +14,8 @@

```

#include <net/ip_fib.h>
#include <linux/inetdevice.h>
#include <linux/in.h>
+#include <linux/skbuff.h>
+#include <linux/ip.h>

```

```

static spinlock_t net_ns_list_lock = SPIN_LOCK_UNLOCKED;

```

@@ -376,14 +378,19 @@ out:

```

    return addr;
}

```

```

-struct net_namespace *net_ns_find_from_dest_addr(u32 daddr)
+struct net_namespace *net_ns_find_from_dest_addr(const struct sk_buff *skb)
{
    struct net_namespace *net_ns = NULL;
    struct net_device *dev;
    struct in_device *in_dev;
+ struct iphdr *iph;
+ __be32 daddr;

```

```

- if (LOOPBACK(daddr))
- return current_net_ns;
+ iph = skb->nh.iph;
+ daddr = iph->daddr;
+
+ if (LOOPBACK(daddr))
+ return skb->net_ns;

```

```

    read_lock(&dev_base_lock);
    rcu_read_lock();

```

--

Subject: [patch 17/20] [Network namespace] For debug purpose only. Add
/sys/kernel/debug/net_ns. Creation of
Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

echo <level> > /sys/kernel/debug/net_ns/start

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

```
fs/debugfs/Makefile | 2
fs/debugfs/net_ns.c | 153 +++++
net/Kconfig          | 4 +
3 files changed, 158 insertions(+), 1 deletion(-)
```

Index: 2.6.19-rc6-mm2/fs/debugfs/Makefile

```
=====
--- 2.6.19-rc6-mm2.orig/fs/debugfs/Makefile
+++ 2.6.19-rc6-mm2/fs/debugfs/Makefile
@@ -1,4 +1,4 @@
debugfs-objs := inode.o file.o
```

```
obj-$(CONFIG_DEBUG_FS) += debugfs.o
```

-

```
+obj-$(CONFIG_NET_NS_DEBUG) += net_ns.o
```

Index: 2.6.19-rc6-mm2/fs/debugfs/net_ns.c

```
=====
--- /dev/null
+++ 2.6.19-rc6-mm2/fs/debugfs/net_ns.c
@@ -0,0 +1,153 @@
+/*
+ * net_ns.c - adds a net_ns/ directory to debug NET namespaces
+ *
+ * Author: Daniel Lezcano <dlezcano@fr.ibm.com>
+ *
+ * This program is free software; you can redistribute it and/or
+ * modify it under the terms of the GNU General Public License as
+ * published by the Free Software Foundation, version 2 of the
+ * License.
+ */
+
+#include <linux/module.h>
```

```

#include <linux/kernel.h>
#include <linux/pagemap.h>
#include <linux/debugfs.h>
#include <linux/sched.h>
#include <linux/netdevice.h>
#include <linux/syscalls.h>
#include <linux/net_namespace.h>
+
+static struct dentry *net_ns_dentry;
+static struct dentry *net_ns_dentry_dev;
+static struct dentry *net_ns_dentry_start;
+
+static ssize_t net_ns_dev_read_file(struct file *file, char __user *user_buf,
+    size_t count, loff_t *ppos)
+{
+ return 0;
+}
+
+static ssize_t net_ns_dev_write_file(struct file *file,
+    const char __user *user_buf,
+    size_t count, loff_t *ppos)
+{
+ return 0;
+}
+
+static int net_ns_dev_open_file(struct inode *inode, struct file *file)
+{
+ return 0;
+}
+
+static int net_ns_start_open_file(struct inode *inode, struct file *file)
+{
+ return 0;
+}
+
+static ssize_t net_ns_start_read_file(struct file *file, char __user *user_buf,
+    size_t count, loff_t *ppos)
+{
+ char c;
+
+ if (*ppos < 0)
+ return -EINVAL;
+ if (*ppos >= count)
+ return 0;
+ if (!count)
+ return 0;
+
+ c = (current_net_ns == &init_net_ns)?'0':'1';

```

```

+
+ if (copy_to_user(user_buf, &c, sizeof(c)))
+ return -EINVAL;
+
+ *ppos += count;
+
+ return count;
+}
+
+int net_ns_start(void);
+
+static ssize_t net_ns_start_write_file(struct file *file,
+    const char __user *user_buf,
+    size_t count, loff_t *ppos)
+{
+ int err;
+ size_t len;
+ const char __user *p;
+ char c;
+ unsigned long flags;
+
+ if (current_net_ns != &init_net_ns)
+ return -EBUSY;
+
+ len = 0;
+ p = user_buf;
+ while (len < count) {
+ if (get_user(c, p++))
+ return -EFAULT;
+ if (c == 0 || c == '\n')
+ break;
+ len++;
+ }
+
+ if (len > 1)
+ return -EINVAL;
+
+ if (copy_from_user(&c, user_buf, sizeof(c)))
+ return -EFAULT;
+
+ if (c != '2' && c != '3')
+ return -EINVAL;
+
+ flags = (c == '2'?NS_NET2:NS_NET3);
+ err = sys_unshare_ns(flags);
+ if (err)
+ return err;
+
+

```

```

+ return count;
+}
+
+static struct file_operations net_ns_dev_fops = {
+    .read =      net_ns_dev_read_file,
+    .write =     net_ns_dev_write_file,
+    .open =      net_ns_dev_open_file,
+};
+
+static struct file_operations net_ns_start_fops = {
+    .read =      net_ns_start_read_file,
+    .write =     net_ns_start_write_file,
+    .open =      net_ns_start_open_file,
+};
+
+static int __init net_ns_init(void)
+{
+    net_ns_dentry = debugfs_create_dir("net_ns", NULL);
+
+    net_ns_dentry_dev = debugfs_create_file("dev", 0666,
+        net_ns_dentry,
+        NULL,
+        &net_ns_dev_fops);
+
+    net_ns_dentry_start = debugfs_create_file("start", 0666,
+        net_ns_dentry,
+        NULL,
+        &net_ns_start_fops);
+
+    return 0;
+}
+
+static void __exit net_ns_exit(void)
+{
+    debugfs_remove(net_ns_dentry_start);
+    debugfs_remove(net_ns_dentry_dev);
+    debugfs_remove(net_ns_dentry);
+}
+
+module_init(net_ns_init);
+module_exit(net_ns_exit);
+
+MODULE_DESCRIPTION("NET namespace debugfs");
+MODULE_AUTHOR("Daniel Lezcano <dlezcano@fr.ibm.com>");
+MODULE_LICENSE("GPL");
Index: 2.6.19-rc6-mm2/net/Kconfig
=====
--- 2.6.19-rc6-mm2.orig/net/Kconfig

```

```
+++ 2.6.19-rc6-mm2/net/Kconfig
@@ -60,6 +60,10 @@ config INET
```

Short answer: say Y.

```
+config NET_NS_DEBUG
+ bool "Debug fs for network namespace"
+ depends on DEBUG_FS && NET_NS
+
+ if INET
+   source "net/ipv4/Kconfig"
+   source "net/ipv6/Kconfig"
```

--

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

Subject: [patch 18/20] [Network namespace] For debug purpose only. Show ref
count for current namespace and l

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

fs/debugfs/net_ns.c | 72 ++++++-----
1 files changed, 71 insertions(+), 1 deletion(-)

Index: 2.6.19-rc6-mm2/fs/debugfs/net_ns.c

=====

--- 2.6.19-rc6-mm2.orig/fs/debugfs/net_ns.c

+++ 2.6.19-rc6-mm2/fs/debugfs/net_ns.c

@@ -21,6 +21,7 @@

```
static struct dentry *net_ns_dentry;
static struct dentry *net_ns_dentry_dev;
static struct dentry *net_ns_dentry_start;
+static struct dentry *net_ns_dentry_info;
```

```
static ssize_t net_ns_dev_read_file(struct file *file, char __user *user_buf,
                                   size_t count, loff_t *ppos)
```

```
@@ -109,6 +110,63 @@ static ssize_t net_ns_start_write_file(s
    return count;
}
```

```

+static int net_ns_info_open_file(struct inode *inode, struct file *file)
+{
+ return 0;
+}
+
+static ssize_t net_ns_info_read_file(struct file *file, char __user *user_buf,
+    size_t count, loff_t *ppos)
+{
+ const unsigned int length = 256;
+ size_t len;
+ char buff[length];
+ char *level;
+ struct net_namespace *net_ns = current_net_ns;
+
+ if (*ppos < 0)
+ return -EINVAL;
+ if (*ppos >= count)
+ return 0;
+ if (!count)
+ return 0;
+
+ switch (net_ns->level) {
+ case NET_NS_LEVEL2:
+ level = "layer 2";
+ break;
+ case NET_NS_LEVEL3:
+ level = "layer 3";
+ break;
+ default:
+ level = "unknown";
+ break;
+ }
+
+ sprintf(buff, "refcnt: %d\nlevel: %s\n",
+ atomic_read(&net_ns->kref.refcount), level);
+
+ len = strlen(buff);
+ if (len > count)
+ len = count;
+
+ if (copy_to_user(user_buf, buff, len))
+ return -EINVAL;
+
+ *ppos += count;
+
+ return count;
+}
+

```

```

+static ssize_t net_ns_info_write_file(struct file *file,
+    const char __user *user_buf,
+    size_t count, loff_t *ppos)
+{
+
+ return -EPERM;
+}
+
+
+static struct file_operations net_ns_dev_fops = {
+    .read =    net_ns_dev_read_file,
+    .write =   net_ns_dev_write_file,
@@ -121,11 +179,17 @@ static struct file_operations net_ns_sta
+    .open =    net_ns_start_open_file,
+};

+static struct file_operations net_ns_info_fops = {
+    .read =    net_ns_info_read_file,
+    .write =   net_ns_info_write_file,
+    .open =    net_ns_info_open_file,
+};
+
+static int __init net_ns_init(void)
+{
+    net_ns_dentry = debugfs_create_dir("net_ns", NULL);

- net_ns_dentry_dev = debugfs_create_file("dev", 0666,
+ net_ns_dentry_dev = debugfs_create_file("dev", 0444,
+    net_ns_dentry,
+    NULL,
+    &net_ns_dev_fops);
@@ -135,11 +199,17 @@ static int __init net_ns_init(void)
+    NULL,
+    &net_ns_start_fops);

+ net_ns_dentry_info = debugfs_create_file("info", 0444,
+    net_ns_dentry,
+    NULL,
+    &net_ns_info_fops);
+
+ return 0;
+}

static void __exit net_ns_exit(void)
{
+ debugfs_remove(net_ns_dentry_info);
+ debugfs_remove(net_ns_dentry_start);
+ debugfs_remove(net_ns_dentry_dev);

```

```
debugfs_remove(net_ns_dentry);
```

--

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

Subject: [patch 19/20] [Network namespace] For debug purpose only. Add the network namespace pointer to the i

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

fs/debugfs/net_ns.c | 28 ++++++++-----
1 files changed, 9 insertions(+), 19 deletions(-)

Index: 2.6.19-rc6-mm2/fs/debugfs/net_ns.c

=====

--- 2.6.19-rc6-mm2.orig/fs/debugfs/net_ns.c

+++ 2.6.19-rc6-mm2/fs/debugfs/net_ns.c

@@ -49,23 +49,7 @@ static int net_ns_start_open_file(struct
static ssize_t net_ns_start_read_file(struct file *file, char __user *user_buf,
size_t count, loff_t *ppos)

```
{  
- char c;  
-  
- if (*ppos < 0)  
- return -EINVAL;  
- if (*ppos >= count)  
- return 0;  
- if (!count)  
- return 0;  
-  
- c = (current_net_ns == &init_net_ns)?'0':'1';  
-  
- if (copy_to_user(user_buf, &c, sizeof(c)))  
- return -EINVAL;  
-  
- *ppos += count;  
-  
- return count;  
+ return 0;  
}
```

```

int net_ns_start(void);
@@ -123,6 +107,7 @@ static ssize_t net_ns_info_read_file(str
    char buff[length];
    char *level;
    struct net_namespace *net_ns = current_net_ns;
+ struct nsproxy *ns = current->nsproxy;

    if (*ppos < 0)
        return -EINVAL;
@@ -143,8 +128,13 @@ static ssize_t net_ns_info_read_file(str
    break;
}

- sprintf(buff,"refcnt: %d\nlevel: %s\n",
- atomic_read(&net_ns->kref.refcount), level);
+ sprintf(buff,
+ "nsproxy: %p\nnsproxy refcnt: %d\nnet_ns: %p\nnet_ns refcnt: %d\nlevel: %s\n",
+ ns,
+ atomic_read(&ns->count),
+ net_ns,
+ atomic_read(&net_ns->kref.refcount),
+ level);

    len = strlen(buff);
    if (len > count)

```

--

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

Subject: [patch 20/20] [Network namespace] For debug purpose, used to unshare the network namespace.

Posted by [Daniel Lezcano](#) on Sun, 10 Dec 2006 21:58:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>

fs/debugfs/net_ns.c | 131 ++++++-----
1 files changed, 127 insertions(+), 4 deletions(-)

Index: 2.6.19-rc6-mm2/fs/debugfs/net_ns.c

=====

```

--- 2.6.19-rc6-mm2.orig/fs/debugfs/net_ns.c
+++ 2.6.19-rc6-mm2/fs/debugfs/net_ns.c
@@ -15,8 +15,10 @@
#include <linux/debugfs.h>
#include <linux/sched.h>
#include <linux/netdevice.h>
+#include <linux/inetdevice.h>
#include <linux/syscalls.h>
#include <linux/net_namespace.h>
+#include <linux/rtnetlink.h>

static struct dentry *net_ns_dentry;
static struct dentry *net_ns_dentry_dev;
@@ -52,8 +54,6 @@ static ssize_t net_ns_start_read_file(st
    return 0;
}

-int net_ns_start(void);
-
static ssize_t net_ns_start_write_file(struct file *file,
    const char __user *user_buf,
    size_t count, loff_t *ppos)
@@ -63,6 +63,8 @@ static ssize_t net_ns_start_write_file(s
    const char __user *p;
    char c;
    unsigned long flags;
+ struct net_namespace *net, *new_net;
+ struct nsproxy *new_nsproxy = NULL, *old_nsproxy = NULL;

    if (current_net_ns != &init_net_ns)
        return -EBUSY;
@@ -87,10 +89,37 @@ static ssize_t net_ns_start_write_file(s
    return -EINVAL;

    flags = (c=='2'?NS_NET2:NS_NET3);
- err = sys_unshare_ns(flags);
+
+ err = unshare_net_ns(flags, &new_net);
    if (err)
        return err;

+ old_nsproxy = current->nsproxy;
+ new_nsproxy = dup_namespaces(old_nsproxy);
+
+ if (!new_nsproxy) {
+     put_net_ns(new_net);
+     task_unlock(current);
+     return -ENOMEM;

```

```

+ }
+
+ task_lock(current);
+
+ if (new_nsproxy) {
+ current->nsproxy = new_nsproxy;
+ new_nsproxy = old_nsproxy;
+ }
+
+ new_net->ns = current->nsproxy;
+ net = current->nsproxy->net_ns;
+ current->nsproxy->net_ns = new_net;
+ new_net = net;
+
+ task_unlock(current);
+
+ put_nsproxy(new_nsproxy);
+ put_net_ns(new_net);
+
+ return count;
+ }

@@ -152,8 +181,102 @@ static ssize_t net_ns_info_write_file(st
    const char __user *user_buf,
    size_t count, loff_t *ppos)
{
+ struct net_namespace *net_ns = current_net_ns;
+ struct net_device *dev;
+ struct in_device *in_dev;
+ struct in_ifaddr **ifap = NULL;
+ struct in_ifaddr *ifa = NULL;
+ char *colon;
+ int err;
+
+ char buff[1024];
+ char *eth, *addr, *s;
+ __be32 address = 0;
+ __be32 p;
+
+ if (!capable(CAP_NET_ADMIN))
+ return -EPERM;
+
+ if (net_ns->level != NET_NS_LEVEL3)
+ return -EPERM;
+
+ if (count > sizeof(buff))
+ return -EINVAL;
+

```

```

+ if (copy_from_user(buff, user_buf, count))
+ return -EFAULT;
+
+ buff[count] = '\0';
+
+     eth = buff;
+     s = strchr(eth, ' ');
+     if (!s)
+         return -EINVAL;
+     *s = 0;
+
+     addr = s + 1;
+     s = strchr(addr, '.');
+     if (!s)
+         return -EINVAL;
+     *s = 0;
+     p = simple_strtoul(addr, NULL, 0);
+     ((char *)&address)[3] = p;
+     addr = s + 1;
+
+     s = strchr(addr, '.');
+     if (!s)
+         return -EINVAL;
+     *s = 0;
+     p = simple_strtoul(addr, NULL, 0);
+     ((char *)&address)[2] = p;
+     addr = s + 1;
+
+     s = strchr(addr, '.');
+     if (!s)
+         return -EINVAL;
+     *s = 0;
+     p = simple_strtoul(addr, NULL, 0);
+     ((char *)&address)[1] = p;
+     addr = s + 1;
+
+     p = simple_strtoul(addr, NULL, 0);
+     ((char *)&address)[0] = p;
+
+ colon = strchr(eth, ':');
+ if (colon)
+     *colon = 0;
+
+     address = htonl(address);
+
+ rtnl_lock();
+
+ err = -ENODEV;

```

```

+ dev = __dev_get_by_name(eth);
+ if (!dev)
+ goto out;
+
+ if (colon)
+ *colon = ':';
+
+ err = -EADDRNOTAVAIL;
+ in_dev = __in_dev_get_rtnl(dev);
+ if (!in_dev)
+ goto out;
+
+ for (ifap = &in_dev->ifa_list; (ifa = *ifap) != NULL;
+      ifap = &ifa->ifa_next)
+ if (!strcmp(eth, ifa->ifa_label) &&
+     address == ifa->ifa_local)
+ break;
+ if (!ifa)
+ goto out;
+
+ ifa->ifa_net_ns = net_ns;

- return -EPERM;
+ err = count;
+out:
+ rtnl_unlock();
+ return err;
}

```

--

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

Subject: Re: [patch 03/20] [Network namespace] Remove useless code
Posted by [Cedric Le Goater](#) on Mon, 11 Dec 2006 15:08:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

dlezcano@fr.ibm.com wrote:
> Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>
>
> ---
>
> net/core/net_namespace.c | 5 -----

```

> 1 files changed, 5 deletions(-)
>
> Index: 2.6.19-rc6-mm2/net/core/net_namespace.c
> =====
> --- 2.6.19-rc6-mm2.orig/net/core/net_namespace.c
> +++ 2.6.19-rc6-mm2/net/core/net_namespace.c
> @@ -128,11 +128,6 @@ void free_net_ns(struct kref *kref)
>  /* taking lock after atomic_dec_and_test is racy */
>  spin_lock_irqsave(&net_ns_list_lock, flags);
>  ns = container_of(kref, struct net_namespace, kref);
>  - if (atomic_read(&ns->kref.refcount) ||
>  -    list_empty(&ns->sibling_list)) {
>  - spin_unlock_irqrestore(&net_ns_list_lock, flags);
>  - return;
>  - }

```

why ?

```

> list_del_init(&ns->sibling_list);
> spin_unlock_irqrestore(&net_ns_list_lock, flags);
> put_net_ns(ns->parent);
>

```

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