Subject: [patch -mm 01/17] net namespace: empty framework
Posted by Cedric Le Goater on Tue, 05 Dec 2006 10:27:53 GMT
View Forum Message <> Reply to Message

From: Cedric Le Goater <clg@fr.ibm.com>

This patch adds an empty net namespace framework with an API
compatible with the other available namespaces.

It doesn't provide any feature by itself but prepares the ground for
work on L2 and L3 network isolation. It also solves patch conflict
issues in nsproxy, which are painful and boring to resolve.

Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>

---
 include/linux/init_task.h | 2 +
 include/linux/net_namespace.h | 49 ++
 include/linux/nsproxy.h | 2 +
 kernel/nsproxy.c | 12 ++++++++++
 net/Kconfig | 8 ++++++
 net/core/Makefile | 2 -
 net/core/net_namespace.c | 41 ++++++++++++++++++++++++++++++++++
 7 files changed, 115 insertions(+), 1 deletion(-)

Index: 2.6.19-rc6-mm2/include/linux/init_task.h
===
--- 2.6.19-rc6-mm2.orig/include/linux/init_task.h
+++ 2.6.19-rc6-mm2/include/linux/init_task.h
@@ -8,6 +8,7 @@
 #include <linux/lockdep.h>
 #include <linux/ipc.h>
 #include <linux/pid_namespace.h>
+#include <linux/net_namespace.h>

 #define INIT_FDTABLE \
 {   \
@@ -78,6 +79,7 @@ extern struct nsproxy init_nsproxy;
  .id = 0,   \
  .uts_ns = &init_uts_ns,   \
  .mnt_ns = NULL,   \
+ INIT_NET_NS(net_ns)   \
  INIT_IPC_NS(ipc_ns)   \
 }

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

```
+++ 2.6.19-rc6-mm2/include/linux/net_namespace.h
@@ -0,0 +1,49 @@
+#ifndef _LINUX_NET_NAMESPACE_H
+#define _LINUX_NET_NAMESPACE_H
+
+#include <linux/kref.h>
+#include <linux/nsproxy.h>
+
+struct net_namespace {
+ struct kref kref;
+};
+
+extern struct net_namespace init_net_ns;
+
+#ifdef CONFIG_NET_NS
+
+#define INIT_NET_NS(net_ns) .net_ns = &init_net_ns,
+
+static inline void get_net_ns(struct net_namespace *ns)
+{
+ kref_get(&ns->kref);
+}
+
+extern int copy_net_ns(int flags, struct task_struct *tsk);
+
+extern void free_net_ns(struct kref *kref);
+
+static inline void put_net_ns(struct net_namespace *ns)
+{
+ kref_put(&ns->kref, free_net_ns);
+}
+
+#else
+
+#define INIT_NET_NS(net_ns)
+
+static inline void get_net_ns(struct net_namespace *ns)
+{
+}
+
+static inline int copy_net_ns(int flags, struct task_struct *tsk)
+{
+ return 0;
+}
+
+static inline void put_net_ns(struct net_namespace *ns)
+{
+}
```

```
+#endif
+
+#endif /* _LINUX_NET_NAMESPACE_H */
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
@@ -8,6 +8,7 @@ struct mnt_namespace;
 struct uts_namespace;
 struct ipc_namespace;
 struct pid_namespace;
+struct net_namespace;

 /*
  * A structure to contain pointers to all per-process
@@ -29,6 +30,7 @@ struct nsproxy {
  struct ipc_namespace *ipc_ns;
  struct mnt_namespace *mnt_ns;
  struct pid_namespace *pid_ns;
+ struct net_namespace *net_ns;
 };
 extern struct nsproxy init_nsproxy;

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
@@ -20,6 +20,7 @@
 #include <linux/mnt_namespace.h>
 #include <linux/utsname.h>
 #include <linux/pid_namespace.h>
+#include <linux/net_namespace.h>

 struct nsproxy init_nsproxy = INIT_NSPROXY(init_nsproxy);

@@ -71,6 +72,8 @@ struct nsproxy *dup_namespaces(struct ns
    get_ipc_ns(ns->ipc_ns);
   if (ns->pid_ns)
    get_pid_ns(ns->pid_ns);
+  if (ns->net_ns)
+   get_net_ns(ns->net_ns);
  }

  return ns;
@@ -118,10 +121,17 @@ int copy_namespaces(int flags, struct ta
  if (err)
   goto out_pid;
```

```
+ err = copy_net_ns(flags, tsk);
+ if (err)
+  goto out_net;
+
 out:
  put_nsproxy(old_ns);
  return err;

+out_net:
+ if (new_ns->pid_ns)
+  put_pid_ns(new_ns->pid_ns);
 out_pid:
  if (new_ns->ipc_ns)
   put_ipc_ns(new_ns->ipc_ns);
@@ -147,5 +157,7 @@ void free_nsproxy(struct nsproxy *ns)
   put_ipc_ns(ns->ipc_ns);
  if (ns->pid_ns)
   put_pid_ns(ns->pid_ns);
+ if (ns->net_ns)
+  put_net_ns(ns->net_ns);
  kfree(ns);
 }
Index: 2.6.19-rc6-mm2/net/core/Makefile
===================================================================
--- 2.6.19-rc6-mm2.orig/net/core/Makefile
+++ 2.6.19-rc6-mm2/net/core/Makefile
@@ -3,7 +3,7 @@
 #

 obj-y := sock.o request_sock.o skbuff.o iovec.o datagram.o stream.o scm.o \
-  gen_stats.o gen_estimator.o
+  gen_stats.o gen_estimator.o net_namespace.o

 obj-$(CONFIG_SYSCTL) += sysctl_net_core.o

Index: 2.6.19-rc6-mm2/net/core/net_namespace.c
===================================================================
--- /dev/null
+++ 2.6.19-rc6-mm2/net/core/net_namespace.c
@@ -0,0 +1,41 @@
+/*
+ *  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/version.h>
+#include <linux/nsproxy.h>
+#include <linux/net_namespace.h>
+
+struct net_namespace init_net_ns = {
+ .kref = {
+  .refcount = ATOMIC_INIT(2),
+ },
+};
+
+#ifdef CONFIG_NET_NS
+
+int copy_net_ns(int flags, struct task_struct *tsk)
+{
+ struct net_namespace *old_ns = tsk->nsproxy->net_ns;
+ int err = 0;
+
+ if (!old_ns)
+  return 0;
+
+ get_net_ns(old_ns);
+ return err;
+}
+
+void free_net_ns(struct kref *kref)
+{
+ struct net_namespace *ns;
+
+ ns = container_of(kref, struct net_namespace, kref);
+ kfree(ns);
+}
+
+#endif /* CONFIG_NET_NS */
Index: 2.6.19-rc6-mm2/net/Kconfig
===================================================================
--- 2.6.19-rc6-mm2.orig/net/Kconfig
+++ 2.6.19-rc6-mm2/net/Kconfig
@@ -67,6 +67,14 @@ source "net/netlabel/Kconfig"

 endif # if INET

+config NET_NS
+ bool "Network Namespaces"
+ help
+   This option enables multiple independent network namespaces,
+   each having own network devices, IP addresses, routes, and so on.
+   If unsure, answer N.
+
```

+
 config NETWORK_SECMARK
  bool "Security Marking"
  help

--

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