Subject: [PATCH 08/16] net: Make socket creation namespace safe.
Posted by ebiederm on Sat, 08 Sep 2007 21:23:01 GMT
View Forum Message <> Reply to Message

This patch passes in the namespace a new socket should be created in
and has the socket code do the appropriate reference counting. By
virtue of this all socket create methods are touched. In addition
the socket create methods are modified so that they will fail if
you attempt to create a socket in a non-default network namespace.

Failing if we attempt to create a socket outside of the default
network namespace ensures that as we incrementally make the network stack
network namespace aware we will not export functionality that someone
has not audited and made certain is network namespace safe.
Allowing us to partially enable network namespaces before all of the
exotic protocols are supported.

Any protocol layers I have missed will fail to compile because I now
pass an extra parameter into the socket creation code.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
drivers/net/pppoe.c | 4 ++--
drivers/net/pppol2tp.c | 4 ++--
drivers/net/pppox.c | 7 +++++--
include/linux/if_pppox.h | 2 +-
include/linux/net.h | 3 ++-
include/net/llc_conn.h | 2 +-
include/net/sock.h | 4 +++-
net/appletalk/ddp.c | 7 +++++--
net/atm/common.c | 4 ++--
net/atm/common.h | 2 +-
net/atm/pvc.c | 7 +++++--
net/atm/svc.c | 11 +++++++----
net/ax25/af_ax25.c | 9 ++++++---
net/bluetooth/af_bluetooth.c | 7 +++++--
net/bluetooth/bnep/sock.c | 4 ++--
net/bluetooth/cmtp/sock.c | 4 ++--
net/bluetooth/hci_sock.c | 4 ++--
net/bluetooth/hidp/sock.c | 4 ++--
net/bluetooth/l2cap.c | 10 +++++-----
net/bluetooth/rfcomm/sock.c | 10 +++++-----
net/bluetooth/sco.c | 10 +++++-----
net/core/sock.c | 6 ++++--
net/decnet/af_decnet.c | 13 +++++++++--
net/econet/af_econet.c | 7 +++++--
net/ipv4/af_inet.c | 7 +++++--
net/ipv6/af_inet6.c | 7 +++++--

```
 net/ipx/af_ipx.c            |    7 +++++--
 net/irda/af_irda.c          |   11 +++++++----
 net/key/af_key.c            |    7 +++++--
 net/llc/af_llc.c            |    7 +++++--
 net/llc/llc_conn.c          |    6 +++---
 net/netlink/af_netlink.c    |   15 +++++++++------
 net/netrom/af_netrom.c      |    9 ++++++---
 net/packet/af_packet.c      |    7 +++++--
 net/rose/af_rose.c          |    9 ++++++---
 net/rxrpc/af_rxrpc.c        |    7 +++++--
 net/sctp/ipv6.c             |    2 +-
 net/sctp/protocol.c         |    2 +-
 net/socket.c                |    9 +++++----
 net/tipc/socket.c           |    9 ++++++---
 net/unix/af_unix.c          |   13 ++++++++-----
 net/x25/af_x25.c            |   13 ++++++++-----
 42 files changed, 182 insertions(+), 110 deletions(-)

diff --git a/drivers/net/pppoe.c b/drivers/net/pppoe.c
index a9b6971..f8bf5fc 100644
--- a/drivers/net/pppoe.c
+++ b/drivers/net/pppoe.c
@@ -477,12 +477,12 @@ static struct proto pppoe_sk_proto = {
  * Initialize a new struct sock.
  *
  **********************************************************************/
-static int pppoe_create(struct socket *sock)
+static int pppoe_create(struct net *net, struct socket *sock)
 {
  int error = -ENOMEM;
  struct sock *sk;

- sk = sk_alloc(PF_PPPOX, GFP_KERNEL, &pppoe_sk_proto, 1);
+ sk = sk_alloc(net, PF_PPPOX, GFP_KERNEL, &pppoe_sk_proto, 1);
  if (!sk)
   goto out;

diff --git a/drivers/net/pppol2tp.c b/drivers/net/pppol2tp.c
index c12e0a8..07d7f5b 100644
--- a/drivers/net/pppol2tp.c
+++ b/drivers/net/pppol2tp.c
@@ -1423,12 +1423,12 @@ static struct proto pppol2tp_sk_proto = {

 /* socket() handler. Initialize a new struct sock.
  */
-static int pppol2tp_create(struct socket *sock)
+static int pppol2tp_create(struct net *net, struct socket *sock)
 {
```

```
  int error = -ENOMEM;
  struct sock *sk;

- sk = sk_alloc(PF_PPPOX, GFP_KERNEL, &pppol2tp_sk_proto, 1);
+ sk = sk_alloc(net, PF_PPPOX, GFP_KERNEL, &pppol2tp_sk_proto, 1);
  if (!sk)
   goto out;

diff --git a/drivers/net/pppox.c b/drivers/net/pppox.c
index 25c52b5..c6898c1 100644
--- a/drivers/net/pppox.c
+++ b/drivers/net/pppox.c
@@ -104,10 +104,13 @@ int pppox_ioctl(struct socket *sock, unsigned int cmd, unsigned long
arg)

 EXPORT_SYMBOL(pppox_ioctl);

-static int pppox_create(struct socket *sock, int protocol)
+static int pppox_create(struct net *net, struct socket *sock, int protocol)
 {
  int rc = -EPROTOTYPE;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  if (protocol < 0 || protocol > PX_MAX_PROTO)
   goto out;

@@ -123,7 +126,7 @@ static int pppox_create(struct socket *sock, int protocol)
      !try_module_get(pppox_protos[protocol]->owner))
   goto out;

- rc = pppox_protos[protocol]->create(sock);
+ rc = pppox_protos[protocol]->create(net, sock);

  module_put(pppox_protos[protocol]->owner);
 out:
diff --git a/include/linux/if_pppox.h b/include/linux/if_pppox.h
index 2565254..43cfc9f 100644
--- a/include/linux/if_pppox.h
+++ b/include/linux/if_pppox.h
@@ -172,7 +172,7 @@ static inline struct sock *sk_pppox(struct pppox_sock *po)
 struct module;

 struct pppox_proto {
- int  (*create)(struct socket *sock);
+ int  (*create)(struct net *net, struct socket *sock);
  int  (*ioctl)(struct socket *sock, unsigned int cmd,
```

```
      unsigned long arg);
  struct module *owner;
diff --git a/include/linux/net.h b/include/linux/net.h
index efc4517..c136abc 100644
--- a/include/linux/net.h
+++ b/include/linux/net.h
@@ -23,6 +23,7 @@

 struct poll_table_struct;
 struct inode;
+struct net;

 #define NPROTO  34  /* should be enough for now.. */

@@ -169,7 +170,7 @@ struct proto_ops {

 struct net_proto_family {
   int  family;
- int  (*create)(struct socket *sock, int protocol);
+ int  (*create)(struct net *net, struct socket *sock, int protocol);
   struct module *owner;
 };

diff --git a/include/net/llc_conn.h b/include/net/llc_conn.h
index 00730d2..e2374e3 100644
--- a/include/net/llc_conn.h
+++ b/include/net/llc_conn.h
@@ -93,7 +93,7 @@ static __inline__ char llc_backlog_type(struct sk_buff *skb)
   return skb->cb[sizeof(skb->cb) - 1];
 }

-extern struct sock *llc_sk_alloc(int family, gfp_t priority,
+extern struct sock *llc_sk_alloc(struct net *net, int family, gfp_t priority,
       struct proto *prot);
 extern void llc_sk_free(struct sock *sk);

diff --git a/include/net/sock.h b/include/net/sock.h
index 253df3f..898413f 100644
--- a/include/net/sock.h
+++ b/include/net/sock.h
@@ -56,6 +56,7 @@
 #include <asm/atomic.h>
 #include <net/dst.h>
 #include <net/checksum.h>
+#include <net/net_namespace.h>

 /*
   * This structure really needs to be cleaned up.
```

```
@@ -777,7 +778,7 @@ extern void FASTCALL(release_sock(struct sock *sk));
     SINGLE_DEPTH_NESTING)
 #define bh_unlock_sock(__sk) spin_unlock(&((__sk)->sk_lock.slock))

-extern struct sock  *sk_alloc(int family,
+extern struct sock  *sk_alloc(struct net *net, int family,
        gfp_t priority,
        struct proto *prot, int zero_it);
 extern void   sk_free(struct sock *sk);
@@ -1006,6 +1007,7 @@ static inline void sock_copy(struct sock *nsk, const struct sock *osk)
 #endif

  memcpy(nsk, osk, osk->sk_prot->obj_size);
+ get_net(nsk->sk_net);
 #ifdef CONFIG_SECURITY_NETWORK
  nsk->sk_security = sptr;
  security_sk_clone(osk, nsk);
diff --git a/net/appletalk/ddp.c b/net/appletalk/ddp.c
index 594b597..fd1d52f 100644
--- a/net/appletalk/ddp.c
+++ b/net/appletalk/ddp.c
@@ -1026,11 +1026,14 @@ static struct proto ddp_proto = {
  * Create a socket. Initialise the socket, blank the addresses
  * set the state.
  */
-static int atalk_create(struct socket *sock, int protocol)
+static int atalk_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  int rc = -ESOCKTNOSUPPORT;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  /*
   * We permit SOCK_DGRAM and RAW is an extension. It is trivial to do
   * and gives you the full ELAP frame. Should be handy for CAP 8)
@@ -1038,7 +1041,7 @@ static int atalk_create(struct socket *sock, int protocol)
  if (sock->type != SOCK_RAW && sock->type != SOCK_DGRAM)
   goto out;
  rc = -ENOMEM;
- sk = sk_alloc(PF_APPLETALK, GFP_KERNEL, &ddp_proto, 1);
+ sk = sk_alloc(net, PF_APPLETALK, GFP_KERNEL, &ddp_proto, 1);
  if (!sk)
   goto out;
  rc = 0;
diff --git a/net/atm/common.c b/net/atm/common.c
index 299ec1e..e166d9e 100644
```

```
--- a/net/atm/common.c
+++ b/net/atm/common.c
@@ -125,7 +125,7 @@ static struct proto vcc_proto = {
  .obj_size = sizeof(struct atm_vcc),
 };

-int vcc_create(struct socket *sock, int protocol, int family)
+int vcc_create(struct net *net, struct socket *sock, int protocol, int family)
 {
  struct sock *sk;
  struct atm_vcc *vcc;
@@ -133,7 +133,7 @@ int vcc_create(struct socket *sock, int protocol, int family)
  sock->sk = NULL;
  if (sock->type == SOCK_STREAM)
   return -EINVAL;
- sk = sk_alloc(family, GFP_KERNEL, &vcc_proto, 1);
+ sk = sk_alloc(net, family, GFP_KERNEL, &vcc_proto, 1);
  if (!sk)
   return -ENOMEM;
  sock_init_data(sock, sk);
diff --git a/net/atm/common.h b/net/atm/common.h
index ad78c9e..16f32c1 100644
--- a/net/atm/common.h
+++ b/net/atm/common.h
@@ -10,7 +10,7 @@
 #include <linux/poll.h> /* for poll_table */


-int vcc_create(struct socket *sock, int protocol, int family);
+int vcc_create(struct net *net, struct socket *sock, int protocol, int family);
 int vcc_release(struct socket *sock);
 int vcc_connect(struct socket *sock, int itf, short vpi, int vci);
 int vcc_recvmsg(struct kiocb *iocb, struct socket *sock, struct msghdr *msg,
diff --git a/net/atm/pvc.c b/net/atm/pvc.c
index 848e6e1..43e8bf5 100644
--- a/net/atm/pvc.c
+++ b/net/atm/pvc.c
@@ -124,10 +124,13 @@ static const struct proto_ops pvc_proto_ops = {
 };


-static int pvc_create(struct socket *sock,int protocol)
+static int pvc_create(struct net *net, struct socket *sock,int protocol)
 {
+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  sock->ops = &pvc_proto_ops;
```

```
- return vcc_create(sock, protocol, PF_ATMPVC);
+ return vcc_create(net, sock, protocol, PF_ATMPVC);
 }


diff --git a/net/atm/svc.c b/net/atm/svc.c
index 53d04c7..daf9a48 100644
--- a/net/atm/svc.c
+++ b/net/atm/svc.c
@@ -25,7 +25,7 @@
 #include "signaling.h"
 #include "addr.h"

-static int svc_create(struct socket *sock,int protocol);
+static int svc_create(struct net *net, struct socket *sock,int protocol);

 /*
  * Note: since all this is still nicely synchronized with the signaling demon,
@@ -326,7 +326,7 @@ static int svc_accept(struct socket *sock,struct socket *newsock,int flags)

  lock_sock(sk);

- error = svc_create(newsock,0);
+ error = svc_create(sk->sk_net, newsock,0);
  if (error)
   goto out;

@@ -627,12 +627,15 @@ static const struct proto_ops svc_proto_ops = {
 };


-static int svc_create(struct socket *sock,int protocol)
+static int svc_create(struct net *net, struct socket *sock,int protocol)
 {
  int error;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  sock->ops = &svc_proto_ops;
- error = vcc_create(sock, protocol, AF_ATMSVC);
+ error = vcc_create(net, sock, protocol, AF_ATMSVC);
  if (error) return error;
  ATM_SD(sock)->local.sas_family = AF_ATMSVC;
  ATM_SD(sock)->remote.sas_family = AF_ATMSVC;
diff --git a/net/ax25/af_ax25.c b/net/ax25/af_ax25.c
index 1d71f85..def6c42 100644
--- a/net/ax25/af_ax25.c
```

```
+++ b/net/ax25/af_ax25.c
@@ -780,11 +780,14 @@ static struct proto ax25_proto = {
  .obj_size = sizeof(struct sock),
 };

-static int ax25_create(struct socket *sock, int protocol)
+static int ax25_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  ax25_cb *ax25;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  switch (sock->type) {
  case SOCK_DGRAM:
   if (protocol == 0 || protocol == PF_AX25)
@@ -830,7 +833,7 @@ static int ax25_create(struct socket *sock, int protocol)
   return -ESOCKTNOSUPPORT;
  }

- if ((sk = sk_alloc(PF_AX25, GFP_ATOMIC, &ax25_proto, 1)) == NULL)
+ if ((sk = sk_alloc(net, PF_AX25, GFP_ATOMIC, &ax25_proto, 1)) == NULL)
   return -ENOMEM;

  ax25 = sk->sk_protinfo = ax25_create_cb();
@@ -855,7 +858,7 @@ struct sock *ax25_make_new(struct sock *osk, struct ax25_dev
*ax25_dev)
  struct sock *sk;
  ax25_cb *ax25, *oax25;

- if ((sk = sk_alloc(PF_AX25, GFP_ATOMIC, osk->sk_prot, 1)) == NULL)
+ if ((sk = sk_alloc(osk->sk_net, PF_AX25, GFP_ATOMIC, osk->sk_prot, 1)) == NULL)
   return NULL;

  if ((ax25 = ax25_create_cb()) == NULL) {
diff --git a/net/bluetooth/af_bluetooth.c b/net/bluetooth/af_bluetooth.c
index d942b94..1220d8a 100644
--- a/net/bluetooth/af_bluetooth.c
+++ b/net/bluetooth/af_bluetooth.c
@@ -95,10 +95,13 @@ int bt_sock_unregister(int proto)
 }
 EXPORT_SYMBOL(bt_sock_unregister);

-static int bt_sock_create(struct socket *sock, int proto)
+static int bt_sock_create(struct net *net, struct socket *sock, int proto)
 {
  int err;
```

```
+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  if (proto < 0 || proto >= BT_MAX_PROTO)
   return -EINVAL;

@@ -113,7 +116,7 @@ static int bt_sock_create(struct socket *sock, int proto)
  read_lock(&bt_proto_lock);

  if (bt_proto[proto] && try_module_get(bt_proto[proto]->owner)) {
-  err = bt_proto[proto]->create(sock, proto);
+  err = bt_proto[proto]->create(net, sock, proto);
   module_put(bt_proto[proto]->owner);
  }

diff --git a/net/bluetooth/bnep/sock.c b/net/bluetooth/bnep/sock.c
index 10292e7..f718965 100644
--- a/net/bluetooth/bnep/sock.c
+++ b/net/bluetooth/bnep/sock.c
@@ -204,7 +204,7 @@ static struct proto bnep_proto = {
  .obj_size = sizeof(struct bt_sock)
 };

-static int bnep_sock_create(struct socket *sock, int protocol)
+static int bnep_sock_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;

@@ -213,7 +213,7 @@ static int bnep_sock_create(struct socket *sock, int protocol)
  if (sock->type != SOCK_RAW)
   return -ESOCKTNOSUPPORT;

- sk = sk_alloc(PF_BLUETOOTH, GFP_ATOMIC, &bnep_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, GFP_ATOMIC, &bnep_proto, 1);
  if (!sk)
   return -ENOMEM;

diff --git a/net/bluetooth/cmtp/sock.c b/net/bluetooth/cmtp/sock.c
index 19be786..cf700c2 100644
--- a/net/bluetooth/cmtp/sock.c
+++ b/net/bluetooth/cmtp/sock.c
@@ -195,7 +195,7 @@ static struct proto cmtp_proto = {
  .obj_size = sizeof(struct bt_sock)
 };

-static int cmtp_sock_create(struct socket *sock, int protocol)
+static int cmtp_sock_create(struct net *net, struct socket *sock, int protocol)
```

```
 {
  struct sock *sk;

@@ -204,7 +204,7 @@ static int cmtp_sock_create(struct socket *sock, int protocol)
  if (sock->type != SOCK_RAW)
   return -ESOCKTNOSUPPORT;

- sk = sk_alloc(PF_BLUETOOTH, GFP_ATOMIC, &cmtp_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, GFP_ATOMIC, &cmtp_proto, 1);
  if (!sk)
   return -ENOMEM;

diff --git a/net/bluetooth/hci_sock.c b/net/bluetooth/hci_sock.c
index 1dae3df..dc2ba7b 100644
--- a/net/bluetooth/hci_sock.c
+++ b/net/bluetooth/hci_sock.c
@@ -618,7 +618,7 @@ static struct proto hci_sk_proto = {
  .obj_size = sizeof(struct hci_pinfo)
 };

-static int hci_sock_create(struct socket *sock, int protocol)
+static int hci_sock_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;

@@ -629,7 +629,7 @@ static int hci_sock_create(struct socket *sock, int protocol)

  sock->ops = &hci_sock_ops;

- sk = sk_alloc(PF_BLUETOOTH, GFP_ATOMIC, &hci_sk_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, GFP_ATOMIC, &hci_sk_proto, 1);
  if (!sk)
   return -ENOMEM;

diff --git a/net/bluetooth/hidp/sock.c b/net/bluetooth/hidp/sock.c
index 0c18525..1de2b6f 100644
--- a/net/bluetooth/hidp/sock.c
+++ b/net/bluetooth/hidp/sock.c
@@ -246,7 +246,7 @@ static struct proto hidp_proto = {
  .obj_size = sizeof(struct bt_sock)
 };

-static int hidp_sock_create(struct socket *sock, int protocol)
+static int hidp_sock_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;

@@ -255,7 +255,7 @@ static int hidp_sock_create(struct socket *sock, int protocol)
```

```
  if (sock->type != SOCK_RAW)
   return -ESOCKTNOSUPPORT;

- sk = sk_alloc(PF_BLUETOOTH, GFP_ATOMIC, &hidp_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, GFP_ATOMIC, &hidp_proto, 1);
  if (!sk)
   return -ENOMEM;

diff --git a/net/bluetooth/l2cap.c b/net/bluetooth/l2cap.c
index c4e4ce4..36ef27b 100644
--- a/net/bluetooth/l2cap.c
+++ b/net/bluetooth/l2cap.c
@@ -518,11 +518,11 @@ static struct proto l2cap_proto = {
  .obj_size = sizeof(struct l2cap_pinfo)
 };

-static struct sock *l2cap_sock_alloc(struct socket *sock, int proto, gfp_t prio)
+static struct sock *l2cap_sock_alloc(struct net *net, struct socket *sock, int proto, gfp_t prio)
 {
  struct sock *sk;

- sk = sk_alloc(PF_BLUETOOTH, prio, &l2cap_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, prio, &l2cap_proto, 1);
  if (!sk)
   return NULL;

@@ -543,7 +543,7 @@ static struct sock *l2cap_sock_alloc(struct socket *sock, int proto, gfp_t
prio)
  return sk;
 }

-static int l2cap_sock_create(struct socket *sock, int protocol)
+static int l2cap_sock_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;

@@ -560,7 +560,7 @@ static int l2cap_sock_create(struct socket *sock, int protocol)

  sock->ops = &l2cap_sock_ops;

- sk = l2cap_sock_alloc(sock, protocol, GFP_ATOMIC);
+ sk = l2cap_sock_alloc(net, sock, protocol, GFP_ATOMIC);
  if (!sk)
   return -ENOMEM;

@@ -1425,7 +1425,7 @@ static inline int l2cap_connect_req(struct l2cap_conn *conn, struct
l2cap_cmd_hd
   goto response;
```

```
  }

- sk = l2cap_sock_alloc(NULL, BTPROTO_L2CAP, GFP_ATOMIC);
+ sk = l2cap_sock_alloc(parent->sk_net, NULL, BTPROTO_L2CAP, GFP_ATOMIC);
  if (!sk)
   goto response;

diff --git a/net/bluetooth/rfcomm/sock.c b/net/bluetooth/rfcomm/sock.c
index 30586ab..266b697 100644
--- a/net/bluetooth/rfcomm/sock.c
+++ b/net/bluetooth/rfcomm/sock.c
@@ -282,12 +282,12 @@ static struct proto rfcomm_proto = {
  .obj_size = sizeof(struct rfcomm_pinfo)
 };

-static struct sock *rfcomm_sock_alloc(struct socket *sock, int proto, gfp_t prio)
+static struct sock *rfcomm_sock_alloc(struct net *net, struct socket *sock, int proto, gfp_t prio)
 {
  struct rfcomm_dlc *d;
  struct sock *sk;

- sk = sk_alloc(PF_BLUETOOTH, prio, &rfcomm_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, prio, &rfcomm_proto, 1);
  if (!sk)
   return NULL;

@@ -323,7 +323,7 @@ static struct sock *rfcomm_sock_alloc(struct socket *sock, int proto, gfp_t
prio
  return sk;
 }

-static int rfcomm_sock_create(struct socket *sock, int protocol)
+static int rfcomm_sock_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;

@@ -336,7 +336,7 @@ static int rfcomm_sock_create(struct socket *sock, int protocol)

  sock->ops = &rfcomm_sock_ops;

- sk = rfcomm_sock_alloc(sock, protocol, GFP_ATOMIC);
+ sk = rfcomm_sock_alloc(net, sock, protocol, GFP_ATOMIC);
  if (!sk)
   return -ENOMEM;

@@ -868,7 +868,7 @@ int rfcomm_connect_ind(struct rfcomm_session *s, u8 channel, struct
rfcomm_dlc *
   goto done;
```

```
 }

- sk = rfcomm_sock_alloc(NULL, BTPROTO_RFCOMM, GFP_ATOMIC);
+ sk = rfcomm_sock_alloc(parent->sk_net, NULL, BTPROTO_RFCOMM, GFP_ATOMIC);
  if (!sk)
   goto done;

diff --git a/net/bluetooth/sco.c b/net/bluetooth/sco.c
index 3f5163e..65b6fb1 100644
--- a/net/bluetooth/sco.c
+++ b/net/bluetooth/sco.c
@@ -414,11 +414,11 @@ static struct proto sco_proto = {
  .obj_size = sizeof(struct sco_pinfo)
 };

-static struct sock *sco_sock_alloc(struct socket *sock, int proto, gfp_t prio)
+static struct sock *sco_sock_alloc(struct net *net, struct socket *sock, int proto, gfp_t prio)
 {
  struct sock *sk;

- sk = sk_alloc(PF_BLUETOOTH, prio, &sco_proto, 1);
+ sk = sk_alloc(net, PF_BLUETOOTH, prio, &sco_proto, 1);
  if (!sk)
   return NULL;

@@ -439,7 +439,7 @@ static struct sock *sco_sock_alloc(struct socket *sock, int proto, gfp_t
prio)
  return sk;
 }

-static int sco_sock_create(struct socket *sock, int protocol)
+static int sco_sock_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;

@@ -452,7 +452,7 @@ static int sco_sock_create(struct socket *sock, int protocol)

  sock->ops = &sco_sock_ops;

- sk = sco_sock_alloc(sock, protocol, GFP_ATOMIC);
+ sk = sco_sock_alloc(net, sock, protocol, GFP_ATOMIC);
  if (!sk)
   return -ENOMEM;

@@ -807,7 +807,7 @@ static void sco_conn_ready(struct sco_conn *conn)

   bh_lock_sock(parent);
```

```
-  sk = sco_sock_alloc(NULL, BTPROTO_SCO, GFP_ATOMIC);
+  sk = sco_sock_alloc(parent->sk_net, NULL, BTPROTO_SCO, GFP_ATOMIC);
   if (!sk) {
    bh_unlock_sock(parent);
    goto done;
diff --git a/net/core/sock.c b/net/core/sock.c
index 3d16a4b..9feec0f 100644
--- a/net/core/sock.c
+++ b/net/core/sock.c
@@ -863,7 +863,7 @@ static inline void sock_lock_init(struct sock *sk)
  * @prot: struct proto associated with this new sock instance
  * @zero_it: if we should zero the newly allocated sock
  */
-struct sock *sk_alloc(int family, gfp_t priority,
+struct sock *sk_alloc(struct net *net, int family, gfp_t priority,
         struct proto *prot, int zero_it)
 {
  struct sock *sk = NULL;
@@ -884,6 +884,7 @@ struct sock *sk_alloc(int family, gfp_t priority,
     */
    sk->sk_prot = sk->sk_prot_creator = prot;
    sock_lock_init(sk);
+   sk->sk_net = get_net(net);
   }

   if (security_sk_alloc(sk, family, priority))
@@ -923,6 +924,7 @@ void sk_free(struct sock *sk)
         __FUNCTION__, atomic_read(&sk->sk_omem_alloc));

  security_sk_free(sk);
+ put_net(sk->sk_net);
  if (sk->sk_prot_creator->slab != NULL)
   kmem_cache_free(sk->sk_prot_creator->slab, sk);
  else
@@ -932,7 +934,7 @@ void sk_free(struct sock *sk)

 struct sock *sk_clone(const struct sock *sk, const gfp_t priority)
 {
- struct sock *newsk = sk_alloc(sk->sk_family, priority, sk->sk_prot, 0);
+ struct sock *newsk = sk_alloc(sk->sk_net, sk->sk_family, priority, sk->sk_prot, 0);

  if (newsk != NULL) {
   struct sk_filter *filter;
diff --git a/net/decnet/af_decnet.c b/net/decnet/af_decnet.c
index 625d595..aca4c49 100644
--- a/net/decnet/af_decnet.c
+++ b/net/decnet/af_decnet.c
@@ -471,10 +471,10 @@ static struct proto dn_proto = {
```

```
  .obj_size  = sizeof(struct dn_sock),
 };

-static struct sock *dn_alloc_sock(struct socket *sock, gfp_t gfp)
+static struct sock *dn_alloc_sock(struct net *net, struct socket *sock, gfp_t gfp)
 {
  struct dn_scp *scp;
- struct sock *sk = sk_alloc(PF_DECnet, gfp, &dn_proto, 1);
+ struct sock *sk = sk_alloc(net, PF_DECnet, gfp, &dn_proto, 1);

  if  (!sk)
   goto out;
@@ -675,10 +675,13 @@ char *dn_addr2asc(__u16 addr, char *buf)



-static int dn_create(struct socket *sock, int protocol)
+static int dn_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  switch(sock->type) {
   case SOCK_SEQPACKET:
    if (protocol != DNPROTO_NSP)
@@ -691,7 +694,7 @@ static int dn_create(struct socket *sock, int protocol)
  }


- if ((sk = dn_alloc_sock(sock, GFP_KERNEL)) == NULL)
+ if ((sk = dn_alloc_sock(net, sock, GFP_KERNEL)) == NULL)
   return -ENOBUFS;

  sk->sk_protocol = protocol;
@@ -1091,7 +1094,7 @@ static int dn_accept(struct socket *sock, struct socket *newsock, int
flags)

  cb = DN_SKB_CB(skb);
  sk->sk_ack_backlog--;
- newsk = dn_alloc_sock(newsock, sk->sk_allocation);
+ newsk = dn_alloc_sock(sk->sk_net, newsock, sk->sk_allocation);
  if (newsk == NULL) {
   release_sock(sk);
   kfree_skb(skb);
diff --git a/net/econet/af_econet.c b/net/econet/af_econet.c
index 35c96bc..a2429db 100644
```

```
--- a/net/econet/af_econet.c
+++ b/net/econet/af_econet.c
@@ -608,12 +608,15 @@ static struct proto econet_proto = {
  * Create an Econet socket
  */

-static int econet_create(struct socket *sock, int protocol)
+static int econet_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct econet_sock *eo;
  int err;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  /* Econet only provides datagram services. */
  if (sock->type != SOCK_DGRAM)
   return -ESOCKTNOSUPPORT;
@@ -621,7 +624,7 @@ static int econet_create(struct socket *sock, int protocol)
  sock->state = SS_UNCONNECTED;

  err = -ENOBUFS;
- sk = sk_alloc(PF_ECONET, GFP_KERNEL, &econet_proto, 1);
+ sk = sk_alloc(net, PF_ECONET, GFP_KERNEL, &econet_proto, 1);
  if (sk == NULL)
   goto out;

diff --git a/net/ipv4/af_inet.c b/net/ipv4/af_inet.c
index e681034..110a19e 100644
--- a/net/ipv4/af_inet.c
+++ b/net/ipv4/af_inet.c
@@ -241,7 +241,7 @@ EXPORT_SYMBOL(build_ehash_secret);
  * Create an inet socket.
  */

-static int inet_create(struct socket *sock, int protocol)
+static int inet_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct list_head *p;
@@ -253,6 +253,9 @@ static int inet_create(struct socket *sock, int protocol)
  int try_loading_module = 0;
  int err;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
```

```
  if (sock->type != SOCK_RAW &&
      sock->type != SOCK_DGRAM &&
      !inet_ehash_secret)
@@ -320,7 +323,7 @@ lookup_protocol:
  BUG_TRAP(answer_prot->slab != NULL);

  err = -ENOBUFS;
- sk = sk_alloc(PF_INET, GFP_KERNEL, answer_prot, 1);
+ sk = sk_alloc(net, PF_INET, GFP_KERNEL, answer_prot, 1);
  if (sk == NULL)
    goto out;

diff --git a/net/ipv6/af_inet6.c b/net/ipv6/af_inet6.c
index b5f9637..21931c8 100644
--- a/net/ipv6/af_inet6.c
+++ b/net/ipv6/af_inet6.c
@@ -81,7 +81,7 @@ static __inline__ struct ipv6_pinfo *inet6_sk_generic(struct sock *sk)
  return (struct ipv6_pinfo *)(((u8 *)sk) + offset);
 }

-static int inet6_create(struct socket *sock, int protocol)
+static int inet6_create(struct net *net, struct socket *sock, int protocol)
 {
  struct inet_sock *inet;
  struct ipv6_pinfo *np;
@@ -94,6 +94,9 @@ static int inet6_create(struct socket *sock, int protocol)
  int try_loading_module = 0;
  int err;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  if (sock->type != SOCK_RAW &&
      sock->type != SOCK_DGRAM &&
      !inet_ehash_secret)
@@ -159,7 +162,7 @@ lookup_protocol:
  BUG_TRAP(answer_prot->slab != NULL);

  err = -ENOBUFS;
- sk = sk_alloc(PF_INET6, GFP_KERNEL, answer_prot, 1);
+ sk = sk_alloc(net, PF_INET6, GFP_KERNEL, answer_prot, 1);
  if (sk == NULL)
    goto out;

diff --git a/net/ipx/af_ipx.c b/net/ipx/af_ipx.c
index 8400525..ee28bab 100644
--- a/net/ipx/af_ipx.c
+++ b/net/ipx/af_ipx.c
```

```
@@ -1360,11 +1360,14 @@ static struct proto ipx_proto = {
  .obj_size = sizeof(struct ipx_sock),
 };

-static int ipx_create(struct socket *sock, int protocol)
+static int ipx_create(struct net *net, struct socket *sock, int protocol)
 {
  int rc = -ESOCKTNOSUPPORT;
  struct sock *sk;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  /*
   * SPX support is not anymore in the kernel sources. If you want to
   * ressurrect it, completing it and making it understand shared skbs,
@@ -1375,7 +1378,7 @@ static int ipx_create(struct socket *sock, int protocol)
   goto out;

  rc = -ENOMEM;
- sk = sk_alloc(PF_IPX, GFP_KERNEL, &ipx_proto, 1);
+ sk = sk_alloc(net, PF_IPX, GFP_KERNEL, &ipx_proto, 1);
  if (!sk)
   goto out;
 #ifdef IPX_REFCNT_DEBUG
diff --git a/net/irda/af_irda.c b/net/irda/af_irda.c
index c80949a..0328ae2 100644
--- a/net/irda/af_irda.c
+++ b/net/irda/af_irda.c
@@ -60,7 +60,7 @@

 #include <net/irda/af_irda.h>

-static int irda_create(struct socket *sock, int protocol);
+static int irda_create(struct net *net, struct socket *sock, int protocol);

 static const struct proto_ops irda_stream_ops;
 static const struct proto_ops irda_seqpacket_ops;
@@ -831,7 +831,7 @@ static int irda_accept(struct socket *sock, struct socket *newsock, int
flags)

  IRDA_DEBUG(2, "%s()\n", __FUNCTION__);

- err = irda_create(newsock, sk->sk_protocol);
+ err = irda_create(sk->sk_net, newsock, sk->sk_protocol);
  if (err)
   return err;
```

```
@@ -1057,13 +1057,16 @@ static struct proto irda_proto = {
  *    Create IrDA socket
  *
  */
-static int irda_create(struct socket *sock, int protocol)
+static int irda_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct irda_sock *self;

  IRDA_DEBUG(2, "%s()\n", __FUNCTION__);

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  /* Check for valid socket type */
  switch (sock->type) {
  case SOCK_STREAM:     /* For TTP connections with SAR disabled */
@@ -1075,7 +1078,7 @@ static int irda_create(struct socket *sock, int protocol)
  }

  /* Allocate networking socket */
- sk = sk_alloc(PF_IRDA, GFP_ATOMIC, &irda_proto, 1);
+ sk = sk_alloc(net, PF_IRDA, GFP_ATOMIC, &irda_proto, 1);
  if (sk == NULL)
   return -ENOMEM;

diff --git a/net/key/af_key.c b/net/key/af_key.c
index c1a9583..ad9d17f 100644
--- a/net/key/af_key.c
+++ b/net/key/af_key.c
@@ -137,11 +137,14 @@ static struct proto key_proto = {
  .obj_size = sizeof(struct pfkey_sock),
 };

-static int pfkey_create(struct socket *sock, int protocol)
+static int pfkey_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  int err;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  if (!capable(CAP_NET_ADMIN))
   return -EPERM;
  if (sock->type != SOCK_RAW)
@@ -150,7 +153,7 @@ static int pfkey_create(struct socket *sock, int protocol)
```

```
   return -EPROTONOSUPPORT;

  err = -ENOMEM;
- sk = sk_alloc(PF_KEY, GFP_KERNEL, &key_proto, 1);
+ sk = sk_alloc(net, PF_KEY, GFP_KERNEL, &key_proto, 1);
  if (sk == NULL)
   goto out;

diff --git a/net/llc/af_llc.c b/net/llc/af_llc.c
index 6b8a103..b482441 100644
--- a/net/llc/af_llc.c
+++ b/net/llc/af_llc.c
@@ -150,14 +150,17 @@ static struct proto llc_proto = {
  * socket type we have available.
  * Returns 0 upon success, negative upon failure.
  */
-static int llc_ui_create(struct socket *sock, int protocol)
+static int llc_ui_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  int rc = -ESOCKTNOSUPPORT;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  if (likely(sock->type == SOCK_DGRAM || sock->type == SOCK_STREAM)) {
   rc = -ENOMEM;
-  sk = llc_sk_alloc(PF_LLC, GFP_KERNEL, &llc_proto);
+  sk = llc_sk_alloc(net, PF_LLC, GFP_KERNEL, &llc_proto);
   if (sk) {
    rc = 0;
    llc_ui_sk_init(sock, sk);
diff --git a/net/llc/llc_conn.c b/net/llc/llc_conn.c
index 3b8cfbe..8ebc276 100644
--- a/net/llc/llc_conn.c
+++ b/net/llc/llc_conn.c
@@ -700,7 +700,7 @@ static struct sock *llc_create_incoming_sock(struct sock *sk,
          struct llc_addr *saddr,
          struct llc_addr *daddr)
 {
- struct sock *newsk = llc_sk_alloc(sk->sk_family, GFP_ATOMIC,
+ struct sock *newsk = llc_sk_alloc(sk->sk_net, sk->sk_family, GFP_ATOMIC,
        sk->sk_prot);
  struct llc_sock *newllc, *llc = llc_sk(sk);

@@ -867,9 +867,9 @@ static void llc_sk_init(struct sock* sk)
  * Allocates a LLC sock and initializes it. Returns the new LLC sock
  * or %NULL if there's no memory available for one
```

```
 */
-struct sock *llc_sk_alloc(int family, gfp_t priority, struct proto *prot)
+struct sock *llc_sk_alloc(struct net *net, int family, gfp_t priority, struct proto *prot)
 {
- struct sock *sk = sk_alloc(family, priority, prot, 1);
+ struct sock *sk = sk_alloc(net, family, priority, prot, 1);

  if (!sk)
   goto out;
diff --git a/net/netlink/af_netlink.c b/net/netlink/af_netlink.c
index 3982f13..406a493 100644
--- a/net/netlink/af_netlink.c
+++ b/net/netlink/af_netlink.c
@@ -384,15 +384,15 @@ static struct proto netlink_proto = {
  .obj_size = sizeof(struct netlink_sock),
 };

-static int __netlink_create(struct socket *sock, struct mutex *cb_mutex,
-       int protocol)
+static int __netlink_create(struct net *net, struct socket *sock,
+       struct mutex *cb_mutex, int protocol)
 {
  struct sock *sk;
  struct netlink_sock *nlk;

  sock->ops = &netlink_ops;

- sk = sk_alloc(PF_NETLINK, GFP_KERNEL, &netlink_proto, 1);
+ sk = sk_alloc(net, PF_NETLINK, GFP_KERNEL, &netlink_proto, 1);
  if (!sk)
   return -ENOMEM;

@@ -412,13 +412,16 @@ static int __netlink_create(struct socket *sock, struct mutex *cb_mutex,
  return 0;
 }

-static int netlink_create(struct socket *sock, int protocol)
+static int netlink_create(struct net *net, struct socket *sock, int protocol)
 {
  struct module *module = NULL;
  struct mutex *cb_mutex;
  struct netlink_sock *nlk;
  int err = 0;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  sock->state = SS_UNCONNECTED;
```

```
  if (sock->type != SOCK_RAW && sock->type != SOCK_DGRAM)
@@ -441,7 +444,7 @@ static int netlink_create(struct socket *sock, int protocol)
  cb_mutex = nl_table[protocol].cb_mutex;
  netlink_unlock_table();

- if ((err = __netlink_create(sock, cb_mutex, protocol)) < 0)
+ if ((err = __netlink_create(net, sock, cb_mutex, protocol)) < 0)
   goto out_module;

  nlk = nlk_sk(sock->sk);
@@ -1318,7 +1321,7 @@ netlink_kernel_create(int unit, unsigned int groups,
  if (sock_create_lite(PF_NETLINK, SOCK_DGRAM, unit, &sock))
   return NULL;

- if (__netlink_create(sock, cb_mutex, unit) < 0)
+ if (__netlink_create(&init_net, sock, cb_mutex, unit) < 0)
   goto out_sock_release;

  if (groups < 32)
diff --git a/net/netrom/af_netrom.c b/net/netrom/af_netrom.c
index 15c8a92..e969d1b 100644
--- a/net/netrom/af_netrom.c
+++ b/net/netrom/af_netrom.c
@@ -409,15 +409,18 @@ static struct proto nr_proto = {
  .obj_size = sizeof(struct nr_sock),
 };

-static int nr_create(struct socket *sock, int protocol)
+static int nr_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct nr_sock *nr;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  if (sock->type != SOCK_SEQPACKET || protocol != 0)
   return -ESOCKTNOSUPPORT;

- if ((sk = sk_alloc(PF_NETROM, GFP_ATOMIC, &nr_proto, 1)) == NULL)
+ if ((sk = sk_alloc(net, PF_NETROM, GFP_ATOMIC, &nr_proto, 1)) == NULL)
   return -ENOMEM;

  nr = nr_sk(sk);
@@ -459,7 +462,7 @@ static struct sock *nr_make_new(struct sock *osk)
  if (osk->sk_type != SOCK_SEQPACKET)
   return NULL;
```

```
- if ((sk = sk_alloc(PF_NETROM, GFP_ATOMIC, osk->sk_prot, 1)) == NULL)
+ if ((sk = sk_alloc(osk->sk_net, PF_NETROM, GFP_ATOMIC, osk->sk_prot, 1)) == NULL)
   return NULL;

  nr = nr_sk(sk);
diff --git a/net/packet/af_packet.c b/net/packet/af_packet.c
index 1eecbd7..72ff099 100644
--- a/net/packet/af_packet.c
+++ b/net/packet/af_packet.c
@@ -978,13 +978,16 @@ static struct proto packet_proto = {
  * Create a packet of type SOCK_PACKET.
  */

-static int packet_create(struct socket *sock, int protocol)
+static int packet_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct packet_sock *po;
  __be16 proto = (__force __be16)protocol; /* weird, but documented */
  int err;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  if (!capable(CAP_NET_RAW))
   return -EPERM;
  if (sock->type != SOCK_DGRAM && sock->type != SOCK_RAW &&
@@ -994,7 +997,7 @@ static int packet_create(struct socket *sock, int protocol)
  sock->state = SS_UNCONNECTED;

  err = -ENOBUFS;
- sk = sk_alloc(PF_PACKET, GFP_KERNEL, &packet_proto, 1);
+ sk = sk_alloc(net, PF_PACKET, GFP_KERNEL, &packet_proto, 1);
  if (sk == NULL)
   goto out;

diff --git a/net/rose/af_rose.c b/net/rose/af_rose.c
index 48319f7..67e06ab 100644
--- a/net/rose/af_rose.c
+++ b/net/rose/af_rose.c
@@ -499,15 +499,18 @@ static struct proto rose_proto = {
  .obj_size = sizeof(struct rose_sock),
 };

-static int rose_create(struct socket *sock, int protocol)
+static int rose_create(struct net *net, struct socket *sock, int protocol)
 {
```

```
  struct sock *sk;
  struct rose_sock *rose;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  if (sock->type != SOCK_SEQPACKET || protocol != 0)
   return -ESOCKTNOSUPPORT;

- if ((sk = sk_alloc(PF_ROSE, GFP_ATOMIC, &rose_proto, 1)) == NULL)
+ if ((sk = sk_alloc(net, PF_ROSE, GFP_ATOMIC, &rose_proto, 1)) == NULL)
   return -ENOMEM;

  rose = rose_sk(sk);
@@ -545,7 +548,7 @@ static struct sock *rose_make_new(struct sock *osk)
  if (osk->sk_type != SOCK_SEQPACKET)
   return NULL;

- if ((sk = sk_alloc(PF_ROSE, GFP_ATOMIC, &rose_proto, 1)) == NULL)
+ if ((sk = sk_alloc(osk->sk_net, PF_ROSE, GFP_ATOMIC, &rose_proto, 1)) == NULL)
   return NULL;

  rose = rose_sk(sk);
diff --git a/net/rxrpc/af_rxrpc.c b/net/rxrpc/af_rxrpc.c
index 122d55d..0803f30 100644
--- a/net/rxrpc/af_rxrpc.c
+++ b/net/rxrpc/af_rxrpc.c
@@ -606,13 +606,16 @@ static unsigned int rxrpc_poll(struct file *file, struct socket *sock,
 /*
  * create an RxRPC socket
  */
-static int rxrpc_create(struct socket *sock, int protocol)
+static int rxrpc_create(struct net *net, struct socket *sock, int protocol)
 {
  struct rxrpc_sock *rx;
  struct sock *sk;

  _enter("%p,%d", sock, protocol);

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  /* we support transport protocol UDP only */
  if (protocol != PF_INET)
   return -EPROTONOSUPPORT;
@@ -623,7 +626,7 @@ static int rxrpc_create(struct socket *sock, int protocol)
  sock->ops = &rxrpc_rpc_ops;
  sock->state = SS_UNCONNECTED;
```

```
- sk = sk_alloc(PF_RXRPC, GFP_KERNEL, &rxrpc_proto, 1);
+ sk = sk_alloc(net, PF_RXRPC, GFP_KERNEL, &rxrpc_proto, 1);
  if (!sk)
   return -ENOMEM;

diff --git a/net/sctp/ipv6.c b/net/sctp/ipv6.c
index cd57a51..5953260 100644
--- a/net/sctp/ipv6.c
+++ b/net/sctp/ipv6.c
@@ -619,7 +619,7 @@ static struct sock *sctp_v6_create_accept_sk(struct sock *sk,
  struct ipv6_pinfo *newnp, *np = inet6_sk(sk);
  struct sctp6_sock *newsctp6sk;

- newsk = sk_alloc(PF_INET6, GFP_KERNEL, sk->sk_prot, 1);
+ newsk = sk_alloc(sk->sk_net, PF_INET6, GFP_KERNEL, sk->sk_prot, 1);
  if (!newsk)
   goto out;

diff --git a/net/sctp/protocol.c b/net/sctp/protocol.c
index a015454..0052ff4 100644
--- a/net/sctp/protocol.c
+++ b/net/sctp/protocol.c
@@ -545,7 +545,7 @@ static struct sock *sctp_v4_create_accept_sk(struct sock *sk,
 {
  struct inet_sock *inet = inet_sk(sk);
  struct inet_sock *newinet;
- struct sock *newsk = sk_alloc(PF_INET, GFP_KERNEL, sk->sk_prot, 1);
+ struct sock *newsk = sk_alloc(sk->sk_net, PF_INET, GFP_KERNEL, sk->sk_prot, 1);

  if (!newsk)
   goto out;
diff --git a/net/socket.c b/net/socket.c
index 7d44453..bf5aba7 100644
--- a/net/socket.c
+++ b/net/socket.c
@@ -84,6 +84,7 @@
 #include <linux/kmod.h>
 #include <linux/audit.h>
 #include <linux/wireless.h>
+#include <linux/nsproxy.h>

 #include <asm/uaccess.h>
 #include <asm/unistd.h>
@@ -1074,7 +1075,7 @@ call_kill:
  return 0;
 }
```

```
-static int __sock_create(int family, int type, int protocol,
+static int __sock_create(struct net *net, int family, int type, int protocol,
     struct socket **res, int kern)
 {
  int err;
@@ -1150,7 +1151,7 @@ static int __sock_create(int family, int type, int protocol,
  /* Now protected by module ref count */
  rcu_read_unlock();

- err = pf->create(sock, protocol);
+ err = pf->create(net, sock, protocol);
  if (err < 0)
   goto out_module_put;

@@ -1189,12 +1190,12 @@ out_release:

 int sock_create(int family, int type, int protocol, struct socket **res)
 {
- return __sock_create(family, type, protocol, res, 0);
+ return __sock_create(current->nsproxy->net_ns, family, type, protocol, res, 0);
 }

 int sock_create_kern(int family, int type, int protocol, struct socket **res)
 {
- return __sock_create(family, type, protocol, res, 1);
+ return __sock_create(&init_net, family, type, protocol, res, 1);
 }

 asmlinkage long sys_socket(int family, int type, int protocol)
diff --git a/net/tipc/socket.c b/net/tipc/socket.c
index 8411017..e36b4b5 100644
--- a/net/tipc/socket.c
+++ b/net/tipc/socket.c
@@ -162,13 +162,16 @@ static void advance_queue(struct tipc_sock *tsock)
  *
  * Returns 0 on success, errno otherwise
  */
-static int tipc_create(struct socket *sock, int protocol)
+static int tipc_create(struct net *net, struct socket *sock, int protocol)
 {
  struct tipc_sock *tsock;
  struct tipc_port *port;
  struct sock *sk;
  u32 ref;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
```

```
  if (unlikely(protocol != 0))
   return -EPROTONOSUPPORT;

@@ -198,7 +201,7 @@ static int tipc_create(struct socket *sock, int protocol)
   return -EPROTOTYPE;
  }

- sk = sk_alloc(AF_TIPC, GFP_KERNEL, &tipc_proto, 1);
+ sk = sk_alloc(net, AF_TIPC, GFP_KERNEL, &tipc_proto, 1);
  if (!sk) {
   tipc_deleteport(ref);
   return -ENOMEM;
@@ -1372,7 +1375,7 @@ static int accept(struct socket *sock, struct socket *newsock, int flags)
  }
  buf = skb_peek(&sock->sk->sk_receive_queue);

- res = tipc_create(newsock, 0);
+ res = tipc_create(sock->sk->sk_net, newsock, 0);
  if (!res) {
   struct tipc_sock *new_tsock = tipc_sk(newsock->sk);
   struct tipc_portid id;
diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
index 2386090..10e7312 100644
--- a/net/unix/af_unix.c
+++ b/net/unix/af_unix.c
@@ -594,7 +594,7 @@ static struct proto unix_proto = {
  */
 static struct lock_class_key af_unix_sk_receive_queue_lock_key;

-static struct sock * unix_create1(struct socket *sock)
+static struct sock * unix_create1(struct net *net, struct socket *sock)
 {
  struct sock *sk = NULL;
  struct unix_sock *u;
@@ -602,7 +602,7 @@ static struct sock * unix_create1(struct socket *sock)
  if (atomic_read(&unix_nr_socks) >= 2*get_max_files())
   goto out;

- sk = sk_alloc(PF_UNIX, GFP_KERNEL, &unix_proto, 1);
+ sk = sk_alloc(net, PF_UNIX, GFP_KERNEL, &unix_proto, 1);
  if (!sk)
   goto out;

@@ -628,8 +628,11 @@ out:
  return sk;
 }

-static int unix_create(struct socket *sock, int protocol)
```

```
+static int unix_create(struct net *net, struct socket *sock, int protocol)
 {
+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  if (protocol && protocol != PF_UNIX)
   return -EPROTONOSUPPORT;

@@ -655,7 +658,7 @@ static int unix_create(struct socket *sock, int protocol)
   return -ESOCKTNOSUPPORT;
  }

- return unix_create1(sock) ? 0 : -ENOMEM;
+ return unix_create1(net, sock) ? 0 : -ENOMEM;
 }

 static int unix_release(struct socket *sock)
@@ -1039,7 +1042,7 @@ static int unix_stream_connect(struct socket *sock, struct sockaddr
*uaddr,
  err = -ENOMEM;

  /* create new sock for complete connection */
- newsk = unix_create1(NULL);
+ newsk = unix_create1(sk->sk_net, NULL);
  if (newsk == NULL)
   goto out;

diff --git a/net/x25/af_x25.c b/net/x25/af_x25.c
index 479927c..2e99315 100644
--- a/net/x25/af_x25.c
+++ b/net/x25/af_x25.c
@@ -466,10 +466,10 @@ static struct proto x25_proto = {
  .obj_size = sizeof(struct x25_sock),
 };

-static struct sock *x25_alloc_socket(void)
+static struct sock *x25_alloc_socket(struct net *net)
 {
  struct x25_sock *x25;
- struct sock *sk = sk_alloc(AF_X25, GFP_ATOMIC, &x25_proto, 1);
+ struct sock *sk = sk_alloc(net, AF_X25, GFP_ATOMIC, &x25_proto, 1);

  if (!sk)
   goto out;
@@ -485,17 +485,20 @@ out:
  return sk;
 }
```

```
-static int x25_create(struct socket *sock, int protocol)
+static int x25_create(struct net *net, struct socket *sock, int protocol)
 {
  struct sock *sk;
  struct x25_sock *x25;
  int rc = -ESOCKTNOSUPPORT;

+ if (net != &init_net)
+  return -EAFNOSUPPORT;
+
  if (sock->type != SOCK_SEQPACKET || protocol)
   goto out;

  rc = -ENOMEM;
- if ((sk = x25_alloc_socket()) == NULL)
+ if ((sk = x25_alloc_socket(net)) == NULL)
   goto out;

  x25 = x25_sk(sk);
@@ -543,7 +546,7 @@ static struct sock *x25_make_new(struct sock *osk)
  if (osk->sk_type != SOCK_SEQPACKET)
   goto out;

- if ((sk = x25_alloc_socket()) == NULL)
+ if ((sk = x25_alloc_socket(osk->sk_net)) == NULL)
   goto out;

  x25 = x25_sk(sk);
--
1.5.3.rc6.17.g1911
```