Subject: [PATCH (resend)] Don't create tunnels with '%' in name.
Posted by Pavel Emelianov on Wed, 27 Feb 2008 07:44:33 GMT
View Forum Message <> Reply to Message

Four tunnel drivers (ip_gre, ipip, ip6_tunnel and sit) can
receive a pre-defined name for a device from the userspace.
Since these drivers call the register_netdevice() (rtnl_lock,
is held), which does _not_ generate the device's name, this
name may contain a '%' character.

Not sure how bad is this to have a device with a '%' in its
name, but all the other places either use the register_netdev(),
which call the dev_alloc_name(), or explicitly call the
dev_alloc_name() before registering, i.e. do not allow for
such names.

This had to be prior to the commit 34cc7b, but I forgot to
number the patches and this one got lost, sorry.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```
diff --git a/net/ipv4/ip_gre.c b/net/ipv4/ip_gre.c
index 906cb1a..e7821ba 100644
--- a/net/ipv4/ip_gre.c
+++ b/net/ipv4/ip_gre.c
@@ -266,20 +266,24 @@ static struct ip_tunnel * ipgre_tunnel_locate(struct ip_tunnel_parm
*parms, int
  if (!dev)
    return NULL;

+ if (strchr(name, '%')) {
+  if (dev_alloc_name(dev, name) < 0)
+   goto failed_free;
+ }
+
  dev->init = ipgre_tunnel_init;
  nt = netdev_priv(dev);
  nt->parms = *parms;

- if (register_netdevice(dev) < 0) {
-  free_netdev(dev);
-  goto failed;
- }
+ if (register_netdevice(dev) < 0)
+  goto failed_free;
```

```
  dev_hold(dev);
  ipgre_tunnel_link(nt);
  return nt;

-failed:
+failed_free:
+ free_netdev(dev);
  return NULL;
 }

diff --git a/net/ipv4/ipip.c b/net/ipv4/ipip.c
index e77e3b8..dbaed69 100644
--- a/net/ipv4/ipip.c
+++ b/net/ipv4/ipip.c
@@ -228,20 +228,24 @@ static struct ip_tunnel * ipip_tunnel_locate(struct ip_tunnel_parm
*parms, int c
  if (dev == NULL)
   return NULL;

+ if (strchr(name, '%')) {
+  if (dev_alloc_name(dev, name) < 0)
+   goto failed_free;
+ }
+
  nt = netdev_priv(dev);
  dev->init = ipip_tunnel_init;
  nt->parms = *parms;

- if (register_netdevice(dev) < 0) {
-  free_netdev(dev);
-  goto failed;
- }
+ if (register_netdevice(dev) < 0)
+  goto failed_free;

  dev_hold(dev);
  ipip_tunnel_link(nt);
  return nt;

-failed:
+failed_free:
+ free_netdev(dev);
  return NULL;
 }

diff --git a/net/ipv6/ip6_tunnel.c b/net/ipv6/ip6_tunnel.c
index 2a124e9..78f4388 100644
--- a/net/ipv6/ip6_tunnel.c
```

```
+++ b/net/ipv6/ip6_tunnel.c
@@ -238,17 +238,24 @@ static struct ip6_tnl *ip6_tnl_create(struct ip6_tnl_parm *p)
  if (dev == NULL)
   goto failed;

+ if (strchr(name, '%')) {
+  if (dev_alloc_name(dev, name) < 0)
+   goto failed_free;
+ }
+
  t = netdev_priv(dev);
  dev->init = ip6_tnl_dev_init;
  t->parms = *p;

- if ((err = register_netdevice(dev)) < 0) {
-  free_netdev(dev);
-  goto failed;
- }
+ if ((err = register_netdevice(dev)) < 0)
+  goto failed_free;
+
  dev_hold(dev);
  ip6_tnl_link(t);
  return t;
+
+failed_free:
+ free_netdev(dev);
 failed:
  return NULL;
 }
diff --git a/net/ipv6/sit.c b/net/ipv6/sit.c
index dde7801..1656c00 100644
--- a/net/ipv6/sit.c
+++ b/net/ipv6/sit.c
@@ -171,6 +171,11 @@ static struct ip_tunnel * ipip6_tunnel_locate(struct ip_tunnel_parm
*parms, int
  if (dev == NULL)
   return NULL;

+ if (strchr(name, '%')) {
+  if (dev_alloc_name(dev, name) < 0)
+   goto failed_free;
+ }
+
  nt = netdev_priv(dev);
  dev->init = ipip6_tunnel_init;
  nt->parms = *parms;
@@ -178,16 +183,16 @@ static struct ip_tunnel * ipip6_tunnel_locate(struct ip_tunnel_parm
```

```
*parms, int
  if (parms->i_flags & SIT_ISATAP)
   dev->priv_flags |= IFF_ISATAP;

- if (register_netdevice(dev) < 0) {
-  free_netdev(dev);
-  goto failed;
- }
+ if (register_netdevice(dev) < 0)
+  goto failed_free;

  dev_hold(dev);

  ipip6_tunnel_link(nt);
  return nt;

+failed_free:
+ free_netdev(dev);
 failed:
  return NULL;
 }
```