
Subject: [PATCH net-2.6.25 8/10][NETNS][FRAGS]: Isolate the secret interval from namespaces.

Posted by [Pavel Emelianov](#) on Tue, 22 Jan 2008 14:07:08 GMT

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

Since we have one hashtable to lookup the fragment, having different secret_interval-s for hash rebuild doesn't make sense, so move this one to inet_frags.

The inet_frags_ctl becomes empty after this, so remove it. The appropriate ctl table is kept read-only in namespaces.

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

```
---
include/net/inet_frag.h          | 6 +-----
include/net/netns/ipv6.h         | 1 -
net/ipv4/inet_fragment.c         | 4 ++--
net/ipv4/ip_fragment.c           | 8 ++-----
net/ipv6/netfilter/nf_conntrack_reasm.c | 6 +-----
net/ipv6/reassembly.c            | 6 ++-----
6 files changed, 8 insertions(+), 23 deletions(-)
```

```
diff --git a/include/net/inet_frag.h b/include/net/inet_frag.h
```

```
index de41359..1917fbe 100644
```

```
--- a/include/net/inet_frag.h
```

```
+++ b/include/net/inet_frag.h
```

```
@@ -31,18 +31,14 @@ struct inet_frag_queue {
```

```
#define INETFRAGS_HASHSZ 64
```

```
-struct inet_frags_ctl {
```

```
- int secret_interval;
```

```
-};
```

```
-
```

```
struct inet_frags {
```

```
    struct list_head lru_list;
```

```
    struct hlist_head hash[INETFRAGS_HASHSZ];
```

```
    rwlock_t lock;
```

```
    u32 rnd;
```

```
    int qsize;
```

```
+ int secret_interval;
```

```
    struct timer_list secret_timer;
```

```
- struct inet_frags_ctl *ctl;
```

```
    unsigned int (*hashfn)(struct inet_frag_queue *);
```

```
    void (*constructor)(struct inet_frag_queue *q,
```

```
diff --git a/include/net/netns/ipv6.h b/include/net/netns/ipv6.h
```

```

index 87ab56a..187c424 100644
--- a/include/net/netns/ipv6.h
+++ b/include/net/netns/ipv6.h
@@ -14,7 +14,6 @@ struct netns_sysctl_ipv6 {
    struct ctl_table_header *table;
    struct ctl_table_header *frags_hdr;
#ifdef
- struct inet_fragments_ctl frags;
    int bindv6only;
    int flush_delay;
    int ip6_rt_max_size;
diff --git a/net/ipv4/inet_fragment.c b/net/ipv4/inet_fragment.c
index 5ab399c..fcf5252 100644
--- a/net/ipv4/inet_fragment.c
+++ b/net/ipv4/inet_fragment.c
@@ -47,7 +47,7 @@ static void inet_frag_secret_rebuild(unsigned long dummy)
    }
    write_unlock(&f->lock);

- mod_timer(&f->secret_timer, now + f->ctl->secret_interval);
+ mod_timer(&f->secret_timer, now + f->secret_interval);
}

void inet_fragments_init(struct inet_fragments *f)
@@ -65,7 +65,7 @@ void inet_fragments_init(struct inet_fragments *f)

    setup_timer(&f->secret_timer, inet_frag_secret_rebuild,
        (unsigned long)f);
- f->secret_timer.expires = jiffies + f->ctl->secret_interval;
+ f->secret_timer.expires = jiffies + f->secret_interval;
    add_timer(&f->secret_timer);
}
EXPORT_SYMBOL(inet_fragments_init);
diff --git a/net/ipv4/ip_fragment.c b/net/ipv4/ip_fragment.c
index 80c2c19..00646ed 100644
--- a/net/ipv4/ip_fragment.c
+++ b/net/ipv4/ip_fragment.c
@@ -74,10 +74,6 @@ struct ipq {
    struct inet_peer *peer;
};

-static struct inet_fragments_ctl ip4_fragments_ctl __read_mostly = {
- .secret_interval = 10 * 60 * HZ,
-};
-
static struct inet_fragments ip4_fragments;

int ip_frag_nqueues(struct net *net)

```

```

@@ -627,7 +623,7 @@ static struct ctl_table ip4_frgs_ctl_table[] = {
{
    .ctl_name = NET_IPV4_IPFRAG_SECRET_INTERVAL,
    .procname = "ipfrag_secret_interval",
-   .data = &ip4_frgs_ctl.secret_interval,
+   .data = &ip4_frgs.secret_interval,
    .maxlen = sizeof(int),
    .mode = 0644,
    .proc_handler = &proc_dointvec_jiffies,
@@ -720,7 +716,6 @@ static int ipv4_frgs_init_net(struct net *net)
void __init ipfrag_init(void)
{
    ipv4_frgs_init_net(&init_net);
-   ip4_frgs_ctl = &ip4_frgs_ctl;
    ip4_frgs.hashfn = ip4_hashfn;
    ip4_frgs.constructor = ip4_frag_init;
    ip4_frgs.destructor = ip4_frag_free;
@@ -728,6 +723,7 @@ void __init ipfrag_init(void)
    ip4_frgs.qsize = sizeof(struct ipq);
    ip4_frgs.match = ip4_frag_match;
    ip4_frgs.frag_expire = ip_expire;
+   ip4_frgs.secret_interval = 10 * 60 * HZ;
    inet_frgs_init(&ip4_frgs);
}

```

```

diff --git a/net/ipv6/netfilter/nf_conntrack_reasm.c b/net/ipv6/netfilter/nf_conntrack_reasm.c
index c75ac17..6eed991 100644

```

```

--- a/net/ipv6/netfilter/nf_conntrack_reasm.c

```

```

+++ b/net/ipv6/netfilter/nf_conntrack_reasm.c

```

```

@@ -70,10 +70,6 @@ struct nf_ct_frag6_queue
    __u16  nhoffset;
};

```

```

-static struct inet_frgs_ctl nf_frgs_ctl __read_mostly = {
-   .secret_interval = 10 * 60 * HZ,
-};
-

```

```

static struct inet_frgs nf_frgs;
static struct netns_frgs nf_init_frgs;

```

```

@@ -701,7 +697,6 @@ int nf_ct_frag6_kfree_frgs(struct sk_buff *skb)

```

```

int nf_ct_frag6_init(void)
{
-   nf_frgs_ctl = &nf_frgs_ctl;
    nf_frgs.hashfn = nf_hashfn;
    nf_frgs.constructor = ip6_frag_init;
    nf_frgs.destructor = NULL;
}

```

```

@@ -709,6 +704,7 @@ int nf_ct_frag6_init(void)
    nf_frags.qsize = sizeof(struct nf_ct_frag6_queue);
    nf_frags.match = ip6_frag_match;
    nf_frags.frag_expire = nf_ct_frag6_expire;
+ nf_frags.secret_interval = 10 * 60 * HZ;
    nf_init_frags.timeout = IPV6_FRAG_TIMEOUT;
    nf_init_frags.high_thresh = 256 * 1024;
    nf_init_frags.low_thresh = 192 * 1024;
diff --git a/net/ipv6/reassembly.c b/net/ipv6/reassembly.c
index 85f3fa3..8520700 100644
--- a/net/ipv6/reassembly.c
+++ b/net/ipv6/reassembly.c
@@ -658,7 +658,7 @@ static struct ctl_table ip6_frags_ctl_table[] = {
 {
    .ctl_name = NET_IPV6_IP6FRAG_SECRET_INTERVAL,
    .procname = "ip6frag_secret_interval",
-   .data = &init_net.ipv6.sysctl.frags.secret_interval,
+   .data = &ip6_frags.secret_interval,
    .maxlen = sizeof(int),
    .mode = 0644,
    .proc_handler = &proc_dointvec_jiffies,
@@ -719,12 +719,9 @@ static inline void ip6_frags_sysctl_unregister(struct net *net)

static int ipv6_frags_init_net(struct net *net)
{
- ip6_frags.ctl = &net->ipv6.sysctl.frags;
-
    net->ipv6.frags.high_thresh = 256 * 1024;
    net->ipv6.frags.low_thresh = 192 * 1024;
    net->ipv6.frags.timeout = IPV6_FRAG_TIMEOUT;
- net->ipv6.sysctl.frags.secret_interval = 10 * 60 * HZ;

    inet_frags_init_net(&net->ipv6.frags);

@@ -748,6 +745,7 @@ int __init ipv6_frag_init(void)
    ip6_frags.qsize = sizeof(struct frag_queue);
    ip6_frags.match = ip6_frag_match;
    ip6_frags.frag_expire = ip6_frag_expire;
+ ip6_frags.secret_interval = 10 * 60 * HZ;
    inet_frags_init(&ip6_frags);
out:
    return ret;
--
1.5.3.4

```
