
Subject: [PATCH 1/2][INET] Fix potential kfree on vmalloc-ed area of request_sock_queue

Posted by [Pavel Emelianov](#) on Wed, 14 Nov 2007 18:08:29 GMT

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

The request_sock_queue's listen_opt is either vmalloc-ed or kmalloc-ed depending on the number of table entries. Thus it is expected to be handled properly on free, which is done in the reqsk_queue_destroy().

However the error path in inet_csk_listen_start() calls the lite version of reqsk_queue_destroy, called __reqsk_queue_destroy, which calls the kfree unconditionally.

Fix this and move the __reqsk_queue_destroy into a .c file as it looks too big to be inline.

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

diff --git a/include/net/request_sock.h b/include/net/request_sock.h

index 7aed02c..0a954ee 100644

--- a/include/net/request_sock.h

+++ b/include/net/request_sock.h

@@ -136,11 +136,7 @@ static inline struct listen_sock *reqsk_queue_yank_listen_sk(struct request_sock
 return lopt;
}

-static inline void __reqsk_queue_destroy(struct request_sock_queue *queue)

-{
- kfree(reqsk_queue_yank_listen_sk(queue));
-}
-

+extern void __reqsk_queue_destroy(struct request_sock_queue *queue);

extern void reqsk_queue_destroy(struct request_sock_queue *queue);

static inline struct request_sock *

diff --git a/net/core/request_sock.c b/net/core/request_sock.c

index 5f0818d..12a15f5 100644

--- a/net/core/request_sock.c

+++ b/net/core/request_sock.c

@@ -71,6 +71,20 @@ int reqsk_queue_alloc(struct request_sock_queue *queue,

EXPORT_SYMBOL(reqsk_queue_alloc);

+void __reqsk_queue_destroy(struct request_sock_queue *queue)

```

+{
+ struct listen_sock *lopt = reqsk_queue_yank_listen_sk(queue);
+ size_t lopt_size = sizeof(struct listen_sock) +
+ lopt->nr_table_entries * sizeof(struct request_sock *);
+
+ if (lopt_size > PAGE_SIZE)
+ vfree(lopt);
+ else
+ kfree(lopt);
+}
+
+EXPORT_SYMBOL(__reqsk_queue_destroy);
+
+void reqsk_queue_destroy(struct request_sock_queue *queue)
+{
+ /* make all the listen_opt local to us */

```

Subject: Re: [PATCH 1/2][INET] Fix potential kfree on vmalloc-ed area of request_sock_queue

Posted by [Eric Dumazet](#) on Wed, 14 Nov 2007 19:42:38 GMT

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

On Wed, 14 Nov 2007 21:08:29 +0300

Pavel Emelyanov <xemul@openvz.org> wrote:

```

> The request_sock_queue's listen_opt is either vmalloc-ed or
> kmalloc-ed depending on the number of table entries. Thus it
> is expected to be handled properly on free, which is done in
> the reqsk_queue_destroy().
>
> However the error path in inet_csk_listen_start() calls
> the lite version of reqsk_queue_destroy, called
> __reqsk_queue_destroy, which calls the kfree unconditionally.
>
> Fix this and move the __reqsk_queue_destroy into a .c file as
> it looks too big to be inline.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
>
> ---
>
> +void __reqsk_queue_destroy(struct request_sock_queue *queue)
> +{
> + struct listen_sock *lopt = reqsk_queue_yank_listen_sk(queue);

```

WARNING : lopt can be NULL here (or else the locking in reqsk_queue_yank_listen_sk() would

be useless ?)

kfree(NULL) was ok, not NULL->nr_table_entries :)

```
> + size_t lopt_size = sizeof(struct listen_sock) +
> + lopt->nr_table_entries * sizeof(struct request_sock *);
> +
> + if (lopt_size > PAGE_SIZE)
> + vfree(lopt);
> + else
> + kfree(lopt);
> +}
```

Subject: Re: [PATCH 1/2][INET] Fix potential kfree on vmalloc-ed area of request_sock_queue

Posted by [davem](#) on Thu, 15 Nov 2007 00:09:05 GMT

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

From: Eric Dumazet <dada1@cosmosbay.com>

Date: Wed, 14 Nov 2007 20:42:38 +0100

> On Wed, 14 Nov 2007 21:08:29 +0300

> Pavel Emelyanov <xemul@openvz.org> wrote:

>

> > The request_sock_queue's listen_opt is either vmalloc-ed or
> > kmalloc-ed depending on the number of table entries. Thus it
> > is expected to be handled properly on free, which is done in
> > the reqsk_queue_destroy().

> >

> > However the error path in inet_csk_listen_start() calls
> > the lite version of reqsk_queue_destroy, called
> > __reqsk_queue_destroy, which calls the kfree unconditionally.

> >

> > Fix this and move the __reqsk_queue_destroy into a .c file as
> > it looks too big to be inline.

> >

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

> >

> > ---

> >

>

> > +void __reqsk_queue_destroy(struct request_sock_queue *queue)

> > +{

> > + struct listen_sock *lopt = reqsk_queue_yank_listen_sk(queue);

>

> WARNING : lopt can be NULL here (or else the locking in reqsk_queue_yank_listen_sk() would be useless ?)

>
> kfree(NULL) was ok, not NULL->nr_table_entries :)

I think for the error recovery case he is trying to fix all of this locking is unnecessary and we know lopt is not NULL.

Pavel can you rework your fix a bit to deal with this?

Thanks a lot.
