Subject: Problem with TCP window too large for TCPRCVBUF still present
Posted by porridge on Thu, 11 Oct 2007 17:46:05 GMT
View Forum Message <> Reply to Message

Hi,

Trying to debug a problem with stalling connections I found this thread:
http://forum.openvz.org/index.php?t=msg&goto=9018
which describes the exact problems I'm still having with 2.6.18-028.18

Has it been fixed in this version? Or do you still need someone to test
this patch?

regards,
--
Marcin Owsiany <marcin@owsiany.pl> http://marcin.owsiany.pl/
GnuPG: 1024D/60F41216 FE67 DA2D 0ACA FC5E 3F75 D6F6 3A0D 8AA0 60F4 1216

"Every program in development at MIT expands until it can read mail."
-- Unknown

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by gblond on Tue, 08 Jan 2008 08:08:05 GMT
View Forum Message <> Reply to Message

On 11 October 2007 21:46:05 Marcin Owsiany wrote:
> Hi,
>
> Trying to debug a problem with stalling connections I found this thread:
> http://forum.openvz.org/index.php?t=msg&goto=9018
> which describes the exact problems I'm still having with 2.6.18-028.18
>
> Has it been fixed in this version? Or do you still need someone to test
> this patch?

Yes, this patch still requires testing.

>
> regards,

--
Thank,
Vitaliy Gusev

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by porridge on Mon, 04 Feb 2008 20:30:13 GMT
View Forum Message <> Reply to Message

Hi Vitaliy,

[Cc'ing to Kirill, who sent the original patch]

On Tue, Jan 08, 2008 at 11:12:49AM +0300, Vitaliy Gusev wrote:
> On 11 October 2007 21:46:05 Marcin Owsiany wrote:
> > Trying to debug a problem with stalling connections I found this thread:
> > http://forum.openvz.org/index.php?t=msg&goto=9018
> > which describes the exact problems I'm still having with 2.6.18-028.18
> >
> > Has it been fixed in this version? Or do you still need someone to test
> > this patch?
>
> Yes, this patch still requires testing.

Can I have the patch as an attachment? The forum mangled the formatting
and patch refuses to apply it..

--
Marcin Owsiany <marcin@owsiany.pl> http://marcin.owsiany.pl/
GnuPG: 1024D/60F41216 FE67 DA2D 0ACA FC5E 3F75 D6F6 3A0D 8AA0 60F4 1216

"Every program in development at MIT expands until it can read mail."
-- Unknown

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by dev on Tue, 05 Feb 2008 08:32:46 GMT
View Forum Message <> Reply to Message

try this one attached plz.

Marcin Owsiany wrote:
> Hi Vitaliy,
>
> [Cc'ing to Kirill, who sent the original patch]
>
> On Tue, Jan 08, 2008 at 11:12:49AM +0300, Vitaliy Gusev wrote:
>> On 11 October 2007 21:46:05 Marcin Owsiany wrote:
>>> Trying to debug a problem with stalling connections I found this thread:
>>> http://forum.openvz.org/index.php?t=msg&goto=9018
>>> which describes the exact problems I'm still having with 2.6.18-028.18
>>>
>>> Has it been fixed in this version? Or do you still need someone to test

>>> this patch?
>> Yes, this patch still requires testing.
>
> Can I have the patch as an attachment? The forum mangled the formatting
> and patch refuses to apply it..
>

--- ./diff.ve7777 2008-02-05 11:25:37.000000000 +0300
+++ ./diff 2008-02-05 11:32:56.000000000 +0300
@@ -1,131 +0,0 @@
-diff --git a/include/net/tcp.h b/include/net/tcp.h
-index 0ff49a5..7e8f200 100644
---- a/include/net/tcp.h
-+++ b/include/net/tcp.h
-@@ -1815,8 +1815,9 @@ static inline int tcp_win_from_space(int
-/* Note: caller must be prepared to deal with negative returns */
-static inline int tcp_space(const struct sock *sk)
-{
-- return tcp_win_from_space(sk->sk_rcvbuf -
-- atomic_read(&sk->sk_rmem_alloc));
-+ int ub_tcp_rcvbuf = (int) sock_bc(sk)->ub_tcp_rcvbuf;
-+ return tcp_win_from_space(min(sk->sk_rcvbuf, ub_tcp_rcvbuf)
-+ - atomic_read(&sk->sk_rmem_alloc));
-}
-
-static inline int tcp_full_space(const struct sock *sk)
-diff --git a/include/ub/beancounter.h b/include/ub/beancounter.h
-index 3d87afa..fc236e8 100644
---- a/include/ub/beancounter.h
-+++ b/include/ub/beancounter.h
-@@ -144,6 +144,8 @@ struct sock_private {
-unsigned long ubp_rmem_thres;
-unsigned long ubp_wmem_pressure;
-unsigned long ubp_maxadvmss;
-+ /* Total size of all advertised receive windows for all tcp sockets */
-+ unsigned long ubp_rcv_wnd;
-unsigned long ubp_rmem_pressure;
-#define UB_RMEM_EXPAND 0
-#define UB_RMEM_KEEP 1
-@@ -177,6 +179,7 @@ #define ub_held_pages ppriv.ubp_held_pa
-struct sock_private spriv;
-#define ub_rmem_thres spriv.ubp_rmem_thres
-#define ub_maxadvmss spriv.ubp_maxadvmss
-+#define ub_rcv_wnd spriv.ubp_rcv_wnd
-#define ub_rmem_pressure spriv.ubp_rmem_pressure
-#define ub_wmem_pressure spriv.ubp_wmem_pressure
-#define ub_tcp_sk_list spriv.ubp_tcp_socks
-diff --git a/include/ub/ub_sk.h b/include/ub/ub_sk.h

-index e65c9ed..02d0137 100644
---- a/include/ub/ub_sk.h
-+++ b/include/ub/ub_sk.h
-@@ -34,6 +34,8 @@ struct sock_beancounter {
-*/
-unsigned long poll_reserv;
-unsigned long forw_space;
-+ unsigned long ub_tcp_rcvbuf;
-+ unsigned long ub_rcv_wnd_old;
-/* fields below are protected by bc spinlock */
-unsigned long ub_waitspc; /* space waiting for */
-unsigned long ub_wcharged;
-diff --git a/kernel/ub/ub_net.c b/kernel/ub/ub_net.c
-index 74d651a..afee710 100644
---- a/kernel/ub/ub_net.c
-+++ b/kernel/ub/ub_net.c
-@@ -420,6 +420,7 @@ static int __sock_charge(struct sock *sk
-
-added_reserv = 0;
-added_forw = 0;
-+ skbc->ub_rcv_wnd_old = 0;
-if (res == UB_NUMTCPSOCK) {
-added_reserv = skb_charge_size(MAX_TCP_HEADER +
-1500 - sizeof(struct iphdr) -
-@@ -439,6 +440,7 @@ static int __sock_charge(struct sock *sk
-added_forw = 0;
-}
-skbc->forw_space = added_forw;
-+ skbc->ub_tcp_rcvbuf = added_forw + SK_STREAM_MEM_QUANTUM;
-}
-spin_unlock_irqrestore(&ub->ub_lock, flags);
-
-@@ -528,6 +530,7 @@ void ub_sock_uncharge(struct sock *sk)
-skbc->ub_wcharged, skbc->ub, skbc->ub->ub_uid);
-skbc->poll_reserv = 0;
-skbc->forw_space = 0;
-+ ub->ub_rcv_wnd -= is_tcp_sock ? tcp_sk(sk)->rcv_wnd : 0;
-spin_unlock_irqrestore(&ub->ub_lock, flags);
-
-uncharge_beancounter_notop(skbc->ub,
-@@ -768,6 +771,44 @@ static void ub_sockrcvbuf_uncharge(struc
-* UB_TCPRCVBUF
-*/
-
-+/*
-+ * UBC TCP window management mechanism.
-+ * Every socket may consume no more than sock_quantum.
-+ * sock_quantum depends on space available and ub_parms[UB_NUMTCPSOCK].held.

```
-    + */
-+static void ub_sock_tcp_update_rcvbuf(struct user_beancounter *ub,
- + struct sock *sk)
-+{
-+ unsigned long allowed;
-+ unsigned long reserved;
-+ unsigned long available;
-+ unsigned long sock_quantum;
-+ struct tcp_opt *tp = tcp_sk(sk);
-+ struct sock_beancounter *skbc;
-+ skbc = sock_bc(sk);
-+
-+ if( ub->ub_parms[UB_NUMTCPSOCK].limit * ub->ub_maxadvmss
- + > ub->ub_parms[UB_TCPRCVBUF].limit) {
- + /* this is defenitly shouldn't happend */
-  + return;
- + }
- + allowed = ub->ub_parms[UB_TCPRCVBUF].barrier;
- + ub->ub_rcv_wnd += (tp->rcv_wnd - skbc->ub_rcv_wnd_old);
- + skbc->ub_rcv_wnd_old = tp->rcv_wnd;
- + reserved = ub->ub_parms[UB_TCPRCVBUF].held + ub->ub_rcv_wnd;
- + available = (allowed < reserved)?
-  + 0:allowed - reserved;
- + sock_quantum = max(allowed / ub->ub_parms[UB_NUMTCPSOCK].held,
-   + ub->ub_maxadvmss);
- + if ( skbc->ub_tcp_rcvbuf > sock_quantum) {
-  + skbc->ub_tcp_rcvbuf = sock_quantum;
-  + } else {
-   + skbc->ub_tcp_rcvbuf += min(sock_quantum - skbc->ub_tcp_rcvbuf,
-     + available);
-   + }
-   +
-    +}
-   +
-    int ub_sock_tcp_chargerecv(struct sock *sk, struct sk_buff *skb,
-      enum ub_severity strict)
-    {
-     @@ -804,6 +845,7 @@ int ub_sock_tcp_chargerecv(struct sock *
-       retval = 0;
-       for (ub = sock_bc(sk)->ub; ub->parent != NULL; ub = ub->parent);
-       spin_lock_irqsave(&ub->ub_lock, flags);
-       + ub_sock_tcp_update_rcvbuf(ub, sk);
-       ub->ub_parms[UB_TCPRCVBUF].held += chargesize;
-       if (ub->ub_parms[UB_TCPRCVBUF].held >
-        ub->ub_parms[UB_TCPRCVBUF].barrier &&
--- ./include/net/tcp.h.ve7777 2007-12-28 18:24:57.000000000 +0300
+++ ./include/net/tcp.h 2008-02-05 11:32:48.000000000 +0300
@@ -971,8 +971,9 @@ static inline int tcp_win_from_space(int
```

```
 /* Note: caller must be prepared to deal with negative returns */
 static inline int tcp_space(const struct sock *sk)
 {
- return tcp_win_from_space(sk->sk_rcvbuf -
-      atomic_read(&sk->sk_rmem_alloc));
+ int ub_tcp_rcvbuf = (int) sock_bc(sk)->ub_tcp_rcvbuf;
+ return tcp_win_from_space(min(sk->sk_rcvbuf, ub_tcp_rcvbuf)
+    - atomic_read(&sk->sk_rmem_alloc));
 }

 static inline int tcp_full_space(const struct sock *sk)
--- ./include/ub/beancounter.h.ve7777 2007-12-28 18:24:56.000000000 +0300
+++ ./include/ub/beancounter.h 2008-02-05 11:32:48.000000000 +0300
@@ -151,6 +151,8 @@ struct sock_private {
  unsigned long  ubp_rmem_thres;
  unsigned long  ubp_wmem_pressure;
  unsigned long  ubp_maxadvmss;
+ /* Total size of all advertised receive windows for all tcp sockets */
+ unsigned long  ubp_rcv_wnd;
  unsigned long  ubp_rmem_pressure;
  int   ubp_tw_count;
 #define UB_RMEM_EXPAND         0
@@ -218,6 +220,7 @@ struct user_beancounter
  struct sock_private spriv;
 #define ub_rmem_thres  spriv.ubp_rmem_thres
 #define ub_maxadvmss  spriv.ubp_maxadvmss
+#define ub_rcv_wnd  spriv.ubp_rcv_wnd
 #define ub_rmem_pressure spriv.ubp_rmem_pressure
 #define ub_wmem_pressure spriv.ubp_wmem_pressure
 #define ub_tcp_sk_list  spriv.ubp_tcp_socks
--- ./include/ub/ub_sk.h.ve7777 2007-12-28 18:24:52.000000000 +0300
+++ ./include/ub/ub_sk.h 2008-02-05 11:32:48.000000000 +0300
@@ -34,6 +34,8 @@ struct sock_beancounter {
   */
  unsigned long          poll_reserv;
  unsigned long  forw_space;
+ unsigned long  ub_tcp_rcvbuf;
+ unsigned long  ub_rcv_wnd_old;
  /* fields below are protected by bc spinlock */
  unsigned long          ub_waitspc;     /* space waiting for */
  unsigned long          ub_wcharged;
--- ./kernel/ub/ub_net.c.ve7777 2007-12-28 18:24:54.000000000 +0300
+++ ./kernel/ub/ub_net.c 2008-02-05 11:32:48.000000000 +0300
@@ -424,6 +424,7 @@ static int __sock_charge(struct sock *sk

  added_reserv = 0;
  added_forw = 0;
+ skbc->ub_rcv_wnd_old = 0;
```

```
  if (res == UB_NUMTCPSOCK) {
   added_reserv = skb_charge_size(MAX_TCP_HEADER +
     1500 - sizeof(struct iphdr) -
@@ -443,6 +444,7 @@ static int __sock_charge(struct sock *sk
    added_forw = 0;
   }
   skbc->forw_space = added_forw;
+  skbc->ub_tcp_rcvbuf = added_forw + SK_STREAM_MEM_QUANTUM;
  }
  spin_unlock_irqrestore(&ub->ub_lock, flags);

@@ -533,6 +535,7 @@ void ub_sock_uncharge(struct sock *sk)
         skbc->ub_wcharged, skbc->ub, skbc->ub->ub_uid);
  skbc->poll_reserv = 0;
  skbc->forw_space = 0;
+ ub->ub_rcv_wnd -= is_tcp_sock ? tcp_sk(sk)->rcv_wnd : 0;
  spin_unlock_irqrestore(&ub->ub_lock, flags);

  uncharge_beancounter_notop(skbc->ub,
@@ -793,6 +796,44 @@ static void ub_sockrcvbuf_uncharge(struc
  * UB_TCPRCVBUF
  */

+/*
+ * UBC TCP window management mechanism.
+ * Every socket may consume no more than sock_quantum.
+ * sock_quantum depends on space available and ub_parms[UB_NUMTCPSOCK].held.
+ */
+static void ub_sock_tcp_update_rcvbuf(struct user_beancounter *ub,
+  struct sock *sk)
+{
+ unsigned long allowed;
+ unsigned long reserved;
+ unsigned long available;
+ unsigned long sock_quantum;
+ struct tcp_opt *tp = tcp_sk(sk);
+ struct sock_beancounter *skbc;
+ skbc = sock_bc(sk);
+
+ if( ub->ub_parms[UB_NUMTCPSOCK].limit * ub->ub_maxadvmss
+    > ub->ub_parms[UB_TCPRCVBUF].limit) {
+  /* this is defenitly shouldn't happend */
+  return;
+ }
+ allowed = ub->ub_parms[UB_TCPRCVBUF].barrier;
+ ub->ub_rcv_wnd += (tp->rcv_wnd - skbc->ub_rcv_wnd_old);
+ skbc->ub_rcv_wnd_old = tp->rcv_wnd;
+ reserved = ub->ub_parms[UB_TCPRCVBUF].held + ub->ub_rcv_wnd;
```

```
+ available = (allowed < reserved)?
+  0:allowed - reserved;
+ sock_quantum = max(allowed / ub->ub_parms[UB_NUMTCPSOCK].held,
+   ub->ub_maxadvmss);
+ if ( skbc->ub_tcp_rcvbuf > sock_quantum) {
+  skbc->ub_tcp_rcvbuf = sock_quantum;
+ } else {
+  skbc->ub_tcp_rcvbuf += min(sock_quantum - skbc->ub_tcp_rcvbuf,
+    available);
+ }
+
+}
+
 int ub_sock_tcp_chargerecv(struct sock *sk, struct sk_buff *skb,
        enum ub_severity strict)
 {
@@ -829,6 +870,7 @@ int ub_sock_tcp_chargerecv(struct sock *
  retval = 0;
  ub = top_beancounter(sock_bc(sk)->ub);
  spin_lock_irqsave(&ub->ub_lock, flags);
+ ub_sock_tcp_update_rcvbuf(ub, sk);
  ub->ub_parms[UB_TCPRCVBUF].held += chargesize;
  if (ub->ub_parms[UB_TCPRCVBUF].held >
    ub->ub_parms[UB_TCPRCVBUF].barrier &&
```

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by porridge on Fri, 14 Mar 2008 16:03:54 GMT
View Forum Message <> Reply to Message

Hi,

Sorry it took so long to try, but that's how life is if you get a
working workaround - little incentive to fix things properly :)

On Tue, Feb 05, 2008 at 11:32:46AM +0300, Kirill Korotaev wrote:
> try this one attached plz.

Unfortunately I get an error:

```
|   CC      kernel/ub/ub_net.o
| mm/mmap.c: In function ‘acct_stack_growth’:
| mm/mmap.c:1546: warning: label ‘fail_sec’ defined but not used
| kernel/ub/ub_net.c: In function ‘ub_sock_tcp_update_rcvbuf’:
| kernel/ub/ub_net.c:795: warning: initialization from incompatible pointer type
| kernel/ub/ub_net.c:805: error: dereferencing pointer to incomplete type
| kernel/ub/ub_net.c:806: error: dereferencing pointer to incomplete type
```

I guess this is because the patch is against a different version. I also had to
tweak the patch by hand so it would apply - looks like top_beancounter() was
introduced after the version I have.

My version is ovz-028.18-deb.patch (i.e. the version which applies to Debian's
2.6.18).

Attached is the ub_net.c file as I have it after applying your tcp window
patch. Just in case it's obvious for you how to fix it to work with 028.18-deb.

regards,
--
Marcin Owsiany <marcin@owsiany.pl> http://marcin.owsiany.pl/
GnuPG: 1024D/60F41216 FE67 DA2D 0ACA FC5E 3F75 D6F6 3A0D 8AA0 60F4 1216

"Every program in development at MIT expands until it can read mail."
-- Unknown

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by gblond on Fri, 14 Mar 2008 16:36:18 GMT
View Forum Message <> Reply to Message

On 14 March 2008 19:03:54 Marcin Owsiany wrote:
> Hi,
>
> Sorry it took so long to try, but that's how life is if you get a
> working workaround - little incentive to fix things properly :)
>
> On Tue, Feb 05, 2008 at 11:32:46AM +0300, Kirill Korotaev wrote:
> > try this one attached plz.
>
> Unfortunately I get an error:
>
> | CC   kernel/ub/ub_net.o
> | mm/mmap.c: In function ‘acct_stack_growth’:
> | mm/mmap.c:1546: warning: label ‘fail_sec’ defined but not used
> | kernel/ub/ub_net.c: In function ‘ub_sock_tcp_update_rcvbuf’:
> | kernel/ub/ub_net.c:795: warning: initialization from incompatible pointer type
> | kernel/ub/ub_net.c:805: error: dereferencing pointer to incomplete type
> | kernel/ub/ub_net.c:806: error: dereferencing pointer to incomplete type
>
> I guess this is because the patch is against a different version. I also had to
> tweak the patch by hand so it would apply - looks like top_beancounter() was

I am wonder as top_beancounter() was introduced at 2007-05-15.
What exactly OpenVZ version do you use?

> introduced after the version I have.
>
> My version is ovz-028.18-deb.patch (i.e. the version which applies to Debian's
> 2.6.18).

>
> Attached is the ub_net.c file as I have it after applying your tcp window
> patch. Just in case it's obvious for you how to fix it to work with 028.18-deb.
>

I didn't get any attachments...

> regards,

--
Thank,
Vitaliy Gusev

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by porridge on Fri, 14 Mar 2008 19:31:50 GMT
View Forum Message <> Reply to Message

On Fri, Mar 14, 2008 at 07:36:18PM +0300, Vitaliy Gusev wrote:
> On 14 March 2008 19:03:54 Marcin Owsiany wrote:
> >
> > Unfortunately I get an error:
> >
> > | CC kernel/ub/ub_net.o
> > | mm/mmap.c: In function ‘acct_stack_growth’:
> > | mm/mmap.c:1546: warning: label ‘fail_sec’ defined but not used
> > | kernel/ub/ub_net.c: In function ‘ub_sock_tcp_update_rcvbuf’:
> > | kernel/ub/ub_net.c:795: warning: initialization from incompatible pointer type
> > | kernel/ub/ub_net.c:805: error: dereferencing pointer to incomplete type
> > | kernel/ub/ub_net.c:806: error: dereferencing pointer to incomplete type
> >
> > I guess this is because the patch is against a different version. I also had to
> > tweak the patch by hand so it would apply - looks like top_beancounter() was
>
> I am wonder as top_beancounter() was introduced at 2007-05-15.
> What exactly OpenVZ version do you use?

It's called diff-ovz-028.18-deb and I took it from the
kernel-patch-openvz debian package maintained by Ola Lindqvist.
You can have a look at the diff, it's almost the same as the one in
upstream directory within

http://ftp.de.debian.org/debian/pool/main/k/kernel-patch-openvz/kernel-patch-openvz_028.18.1etch5.tar.gz

Almost all timestamps in the diff file are around 2007-03-09 17:52

> > Attached is the ub_net.c file as I have it after applying your tcp window
> > patch. Just in case it's obvious for you how to fix it to work with 028.18-deb.
> >
>
> I didn't get any attachments...

Sorry, I'm attaching it now.

--
Marcin Owsiany <marcin@owsiany.pl> http://marcin.owsiany.pl/
GnuPG: 1024D/60F41216 FE67 DA2D 0ACA FC5E 3F75 D6F6 3A0D 8AA0 60F4 1216

"Every program in development at MIT expands until it can read mail."
-- Unknown

File Attachments

1) ub_net.c, downloaded 759 times

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by gblond on Sat, 15 Mar 2008 12:53:21 GMT
View Forum Message <> Reply to Message

On 14 March 2008 22:31:50 Marcin Owsiany wrote:
> On Fri, Mar 14, 2008 at 07:36:18PM +0300, Vitaliy Gusev wrote:
> > On 14 March 2008 19:03:54 Marcin Owsiany wrote:
> > >
> > > Unfortunately I get an error:
> > >
> > > | CC kernel/ub/ub_net.o
> > > | mm/mmap.c: In function ‘acct_stack_growth’:
> > > | mm/mmap.c:1546: warning: label ‘fail_sec’ defined but not used
> > > | kernel/ub/ub_net.c: In function ‘ub_sock_tcp_update_rcvbuf’:
> > > | kernel/ub/ub_net.c:795: warning: initialization from incompatible pointer type
> > > | kernel/ub/ub_net.c:805: error: dereferencing pointer to incomplete type
> > > | kernel/ub/ub_net.c:806: error: dereferencing pointer to incomplete type
> > >
> > > I guess this is because the patch is against a different version. I also had to
> > > tweak the patch by hand so it would apply - looks like top_beancounter() was
> >
> > I am wonder as top_beancounter() was introduced at 2007-05-15.
> > What exactly OpenVZ version do you use?
>

> It's called diff-ovz-028.18-deb and I took it from the
> kernel-patch-openvz debian package maintained by Ola Lindqvist.
> You can have a look at the diff, it's almost the same as the one in
> upstream directory within
>
http://ftp.de.debian.org/debian/pool/main/k/kernel-patch-openvz/kernel-patch-openvz_028.18.1etch5.tar.gz

Patch kernel-patch-openvz_028.18.1 is too old. This patch conforms
with OpenVZ-028stab018.1 version.

I see kernel-patch-openvz_028.51.3d2.tar.gz (that conforms with OpenVZ-028stab051.3)
in this repository. Can you try to use this openvz_028.51.3 patch?

>
> Almost all timestamps in the diff file are around 2007-03-09 17:52
>
> > > Attached is the ub_net.c file as I have it after applying your tcp window
> > > patch. Just in case it's obvious for you how to fix it to work with 028.18-deb.
> > >
> >
> > I didn't get any attachments...
>
> Sorry, I'm attaching it now.
>

--
Thank,
Vitaliy Gusev

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by porridge on Sat, 15 Mar 2008 18:31:08 GMT
View Forum Message <> Reply to Message

[Ola, this is about applying the patch in lenny kernel-patch-openvz to the etch's linux-2.6]

On Sat, Mar 15, 2008 at 03:53:21PM +0300, Vitaliy Gusev wrote:
> Patch kernel-patch-openvz_028.18.1 is too old. This patch conforms
> with OpenVZ-028stab018.1 version.
>
> I see kernel-patch-openvz_028.51.3d2.tar.gz (that conforms with OpenVZ-028stab051.3)
> in this repository. Can you try to use this openvz_028.51.3 patch?

Unfortunately that one does not apply. If I'm reading the .rej file

correctly, it's because the following chunk against net/ipv6/exthdrs.c seems to
be missing the line which is shown as line 236 in
http://git.openvz.org/?p=linux-2.6.18-openvz;a=blame;f=net/ipv6/exthdrs.c;h=38fdd5f4b1c800181b3aea3b47910aadbb1c646e;hb=HEAD

I mean: the line "hdr = (struct ipv6_rt_hdr *) skb->h.raw;" is in debian
kernel source tree, but the following chunk of the 028.51.3d2 patch does
not expect it (there is just an empty line between the switch and preceding
block).

Ola, any ideas what's going on? Is it supposed to apply at all?

```
| @@ -255,6 +255,20 @@ static int ipv6_rthdr_rcv(struct sk_buff
|                 return -1;
|         }
|
| +       switch (hdr->type) {
| +       case IPV6_SRCRT_TYPE_0:
| +               /* Completely disallow routing header type 0 for now, it can be
| +                * made conditional at a later point if needed. Even though the
| +                * code is non functional at the moment, it is left intact to
| +                * allow backporting Mobile IPv6 later on. */
| +               kfree_skb(skb);
| +               return -1;
| +       default:
| +               IP6_INC_STATS_BH(IPSTATS_MIB_INHDRERRORS);
| +               icmpv6_param_prob(skb, ICMPV6_HDR_FIELD, (&hdr->type) - skb->nh.raw);
| +               return -1;
| +       }
| +
|         if (ipv6_addr_is_multicast(&skb->nh.ipv6h->daddr) ||
|             skb->pkt_type != PACKET_HOST) {
|                 IP6_INC_STATS_BH(IPSTATS_MIB_INADDRERRORS);
```

--
Marcin Owsiany <marcin@owsiany.pl> http://marcin.owsiany.pl/
GnuPG: 1024D/60F41216 FE67 DA2D 0ACA FC5E 3F75 D6F6 3A0D 8AA0 60F4 1216

"Every program in development at MIT expands until it can read mail."
-- Unknown

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by opalsys on Sat, 15 Mar 2008 20:36:07 GMT
View Forum Message <> Reply to Message

Hi

Which kernel source version did you use?

I have used:
linux-source-2.6.18 2.6.18.dfsg.1-18etch1
kernel-patch-openvz 028.51.3d2

That applies fine.

Best regards,

// Ola

On Sat, Mar 15, 2008 at 06:31:08PM +0000, Marcin Owsiany wrote:
> [Ola, this is about applying the patch in lenny kernel-patch-openvz to the etch's linux-2.6]
>
> On Sat, Mar 15, 2008 at 03:53:21PM +0300, Vitaliy Gusev wrote:
> > Patch kernel-patch-openvz_028.18.1 is too old. This patch conforms
> > with OpenVZ-028stab018.1 version.
> >
> > I see kernel-patch-openvz_028.51.3d2.tar.gz (that conforms with OpenVZ-028stab051.3)
> > in this repository. Can you try to use this openvz_028.51.3 patch?
>
> Unfortunately that one does not apply. If I'm reading the .rej file
> correctly, it's because the following chunk against net/ipv6/exthdrs.c seems to
> be missing the line which is shown as line 236 in
>
http://git.openvz.org/?p=linux-2.6.18-openvz;a=blame;f=net/ipv6/exthdrs.c;h=38fdd5f4b1c800181b
3aea3b47910aadbb1c646e;hb=HEAD
>
> I mean: the line "hdr = (struct ipv6_rt_hdr *) skb->h.raw;" is in debian
> kernel source tree, but the following chunk of the 028.51.3d2 patch does
> not expect it (there is just an empty line between the switch and preceding
> block).
>
> Ola, any ideas what's going on? Is it supposed to apply at all?
>
>
> | @@ -255,6 +255,20 @@ static int ipv6_rthdr_rcv(struct sk_buff
> | return -1;
> | }
> |
> | + switch (hdr->type) {
> | + case IPV6_SRCRT_TYPE_0:
> | + /* Completely disallow routing header type 0 for now, it can be
> | + * made conditional at a later point if needed. Even though the
> | + * code is non functional at the moment, it is left intact to

```
> | +                * allow backporting Mobile IPv6 later on. */
> | +               kfree_skb(skb);
> | +               return -1;
> | +       default:
> | +               IP6_INC_STATS_BH(IPSTATS_MIB_INHDRERRORS);
> | +               icmpv6_param_prob(skb, ICMPV6_HDR_FIELD, (&hdr->type) - skb->nh.raw);
> | +               return -1;
> | +       }
> | +
> |         if (ipv6_addr_is_multicast(&skb->nh.ipv6h->daddr) ||
> |             skb->pkt_type != PACKET_HOST) {
> |                 IP6_INC_STATS_BH(IPSTATS_MIB_INADDRERRORS);
> 
> 
> --
> Marcin Owsiany <marcin@owsiany.pl>              http://marcin.owsiany.pl/
> GnuPG: 1024D/60F41216  FE67 DA2D 0ACA FC5E 3F75  D6F6 3A0D 8AA0 60F4 1216
> 
> "Every program in development at MIT expands until it can read mail."
>                                                  -- Unknown
> 


--
 --- Ola Lundqvist systemkonsult --- M Sc in IT Engineering ----
/  ola@opalsys.net                  Annebergsslingan 37       \
|  opal@debian.org                   654 65 KARLSTAD          |
|  http://opalsys.net/             Mobile: +46 (0)70-332 1551 |
\  gpg/f.p.: 7090 A92B 18FE 7994 0C36 4FE4 18A1 B1CF 0FE5 3DD9  /
 ----------------------------------------------------------------
```

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by porridge on Thu, 20 Mar 2008 10:20:12 GMT
View Forum Message <> Reply to Message

On Sat, Mar 15, 2008 at 09:36:07PM +0100, Ola Lundqvist wrote:
> Which kernel source version did you use?
>
> I have used:
> linux-source-2.6.18   2.6.18.dfsg.1-18etch1
> kernel-patch-openvz   028.51.3d2
>
> That applies fine.

Right, this is a problem very similar to the one described in
http://bugs.debian.org/470962 (i.e. I'm trying to apply it with max fuzz
level == 1, which does not work).

Now that I at least know what is going on, I'm going to tweak the patch
and try the window size patch.

--
Marcin Owsiany <marcin@owsiany.pl> http://marcin.owsiany.pl/
GnuPG: 1024D/60F41216 FE67 DA2D 0ACA FC5E 3F75 D6F6 3A0D 8AA0 60F4 1216

"Every program in development at MIT expands until it can read mail."
-- Unknown

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by porridge on Thu, 20 Mar 2008 11:56:02 GMT
View Forum Message <> Reply to Message

On Thu, Mar 20, 2008 at 10:20:12AM +0000, Marcin Owsiany wrote:
> Now that I at least know what is going on, I'm going to tweak the patch
> and try the window size patch.

Unfortunately, using the 028.51.3 patch does not help.

kernel/ub/ub_net.c: In function ‘ub_sock_tcp_update_rcvbuf’:
kernel/ub/ub_net.c:811: warning: initialization from incompatible pointer type
kernel/ub/ub_net.c:821: error: dereferencing pointer to incomplete type
kernel/ub/ub_net.c:822: error: dereferencing pointer to incomplete type

Attaching the ub_net.c file it tried to compile.

--
Marcin Owsiany <marcin@owsiany.pl> http://marcin.owsiany.pl/
GnuPG: 1024D/60F41216 FE67 DA2D 0ACA FC5E 3F75 D6F6 3A0D 8AA0 60F4 1216

"Every program in development at MIT expands until it can read mail."
-- Unknown

File Attachments
1) ub_net.c, downloaded 766 times

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by gblond on Thu, 20 Mar 2008 13:19:19 GMT
View Forum Message <> Reply to Message

On 20 March 2008 14:56:02 Marcin Owsiany wrote:
> On Thu, Mar 20, 2008 at 10:20:12AM +0000, Marcin Owsiany wrote:
> > Now that I at least know what is going on, I'm going to tweak the patch
> > and try the window size patch.

>
> Unfortunately, using the 028.51.3 patch does not help.
>
> kernel/ub/ub_net.c: In function ‘ub_sock_tcp_update_rcvbuf’:
> kernel/ub/ub_net.c:811: warning: initialization from incompatible pointer type
> kernel/ub/ub_net.c:821: error: dereferencing pointer to incomplete type
> kernel/ub/ub_net.c:822: error: dereferencing pointer to incomplete type
>
> Attaching the ub_net.c file it tried to compile.
>

Try attached patch again, please.

--
Thank,
Vitaliy Gusev

File Attachments

1) send_ack.patch, downloaded 734 times

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by porridge on Thu, 20 Mar 2008 14:15:25 GMT
View Forum Message <> Reply to Message

On Thu, Mar 20, 2008 at 04:19:19PM +0300, Vitaliy Gusev wrote:
> Try attached patch again, please.

Changing "struct tcp_opt" to "struct tcp_sock" fixed the compilation
issue, thanks.

I also noticed you added another call to ub_sock_tcp_update_rcvbuf(),
here:

> @@ -829,7 +870,9 @@ int ub_sock_tcp_chargerecv(struct sock *sk, struct sk_buff *skb,
> retval = 0;
> ub = top_beancounter(sock_bc(sk)->ub);
> spin_lock_irqsave(&ub->ub_lock, flags);
> + ub_sock_tcp_update_rcvbuf(ub, sk);
> ub->ub_parms[UB_TCPRCVBUF].held += chargesize;
> + ub_sock_tcp_update_rcvbuf(ub, sk);
> if (ub->ub_parms[UB_TCPRCVBUF].held >
> ub->ub_parms[UB_TCPRCVBUF].barrier &&
> strict != UB_FORCE)

I'm not predenting I understand what it's all about, but just wanted to
ask if you really wanted to add this, and it's not just some copy-paste
typo.

Anyway, switching to the newer openvz patch has changed the kernel ABI,
so it will take me a bit more time to sort my package builds and test
the new kernel. This email is just to let you know that it built
successfully.

--
Marcin Owsiany <marcin@owsiany.pl> http://marcin.owsiany.pl/
GnuPG: 1024D/60F41216 FE67 DA2D 0ACA FC5E 3F75 D6F6 3A0D 8AA0 60F4 1216

"Every program in development at MIT expands until it can read mail."
                                        -- Unknown

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by porridge on Sat, 22 Mar 2008 23:17:20 GMT
View Forum Message <> Reply to Message

On Thu, Mar 20, 2008 at 04:19:19PM +0300, Vitaliy Gusev wrote:
> Try attached patch again, please.

When I use a kernel with this patch, I cannot make it deadlock, which is
good, but I also am unable to make the TCP window increase to anything
more than 960 _bytes_, even in VE0, which is obviously very bad (limits
available bandwidh in my test case by about 50%).

As the next step I'm going to build a kernel with the 028.53.3 openvz
patch, but without the window fix patch, to see where the problem lies.

--
Marcin Owsiany <marcin@owsiany.pl> http://marcin.owsiany.pl/
GnuPG: 1024D/60F41216 FE67 DA2D 0ACA FC5E 3F75 D6F6 3A0D 8AA0 60F4 1216

"Every program in development at MIT expands until it can read mail."
                                        -- Unknown

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by porridge on Mon, 24 Mar 2008 15:39:25 GMT
View Forum Message <> Reply to Message

On Sat, Mar 22, 2008 at 11:17:20PM +0000, Marcin Owsiany wrote:
> As the next step I'm going to build a kernel with the 028.53.3 openvz
> patch, but without the window fix patch, to see where the problem lies.

I tried this, and window size is correct (i.e. around 11 KB rather than
below 1 KB) with the 028.53.3 openvz patch, but without the "window size

fix" patch, so the fault must be in the latter.

Please let me know if you have a newer version to be tested.

--
Marcin Owsiany <marcin@owsiany.pl> http://marcin.owsiany.pl/
GnuPG: 1024D/60F41216 FE67 DA2D 0ACA FC5E 3F75 D6F6 3A0D 8AA0 60F4 1216

"Every program in development at MIT expands until it can read mail."
-- Unknown

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by gblond on Tue, 25 Mar 2008 12:32:48 GMT
View Forum Message <> Reply to Message

Hello!

On 24 March 2008 18:39:25 Marcin Owsiany wrote:
> On Sat, Mar 22, 2008 at 11:17:20PM +0000, Marcin Owsiany wrote:
> > As the next step I'm going to build a kernel with the 028.53.3 openvz
> > patch, but without the window fix patch, to see where the problem lies.
>
> I tried this, and window size is correct (i.e. around 11 KB rather than
> below 1 KB) with the 028.53.3 openvz patch, but without the "window size
> fix" patch, so the fault must be in the latter.

Thanks for your testing!

Does the original kernel 028.53.3 still have issue with sending ack?

I see if VE has too small tcprcvbuf (about 30000) then server retransmit
packets to VE. But i can't reproduce a deadlock state.

>
> Please let me know if you have a newer version to be tested.
>

--
Thank,
Vitaliy Gusev

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by porridge on Tue, 25 Mar 2008 12:50:46 GMT
View Forum Message <> Reply to Message

On Tue, Mar 25, 2008 at 03:33:08PM +0300, Vitaliy Gusev wrote:
> Does the original kernel 028.53.3 still have issue with sending ack?

I didn't try.

> I see if VE has too small tcprcvbuf (about 30000) then server retransmit
> packets to VE. But i can't reproduce a deadlock state.

Maybe you need a crappy internet connection to be able to reproduce this
:)

The way I did that recently was:
1) saturate the link (medium-quality 4 Mb ADSL) in both directions from another machine

2) give it a couple of minutes until the transfer rates are steady

3) start downloading a large file in a VE (I'm in UK and used a debian-cd mirror in Australia for that)

4) give it a couple of minutes until the transfer rate is steady (you can observe the current window size in tcpdump - just wait until it stops changing)

5) then stop both transfers on the other machine

6) after some time (10-20 seconds), the testing VE will notice that more bandwidth has become available, and you will notice that the window size will start increasing, to increase the transfer rate

7) for me, at the point the window size has reached about 9KB (it takes just a couple of seconds from the moment the window started increasing), the deadlock occured

--
Marcin Owsiany <marcin@owsiany.pl> http://marcin.owsiany.pl/
GnuPG: 1024D/60F41216 FE67 DA2D 0ACA FC5E 3F75 D6F6 3A0D 8AA0 60F4 1216

"Every program in development at MIT expands until it can read mail."
-- Unknown

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by porridge on Tue, 25 Mar 2008 13:32:24 GMT
View Forum Message <> Reply to Message

On Tue, Mar 25, 2008 at 03:33:08PM +0300, Vitaliy Gusev wrote:
> Does the original kernel 028.53.3 still have issue with sending ack?

I tried now, and it does. I even managed to have it deadlock right at
the beginning of the connection:

192.168.1.177.56433 is a wget in the VE
203.21.20.200.80 is a debian cd mirror HTTP server

As you can see, the packets marked with "*" arrived in inverted order, and that
(and possibly also the subsequent 1771:3219 packet) was enough for the receiver
to lock up.

| 13:15:10.560747 IP 192.168.1.177.56433 > 203.21.20.200.80: S 4210307013:4210307013(0)
win 5840 <mss 1460,sackOK,timestamp 4452410 0,nop,wscale 4>
| 13:15:10.983733 IP 203.21.20.200.80 > 192.168.1.177.56433: S 3826526567:3826526567(0)
ack 4210307014 win 5792 <mss 1460,sackOK,timestamp 3951406805 4452410,nop,wscale 2>
| 13:15:10.983815 IP 192.168.1.177.56433 > 203.21.20.200.80: . ack 1 win 365
<nop,nop,timestamp 4452516 3951406805>
| 13:15:10.983919 IP 192.168.1.177.56433 > 203.21.20.200.80: P 1:164(163) ack 1 win 365
<nop,nop,timestamp 4452516 3951406805>
| 13:15:11.329076 IP 203.21.20.200.80 > 192.168.1.177.56433: . ack 164 win 1716
<nop,nop,timestamp 3951407263 4452516>
* 13:15:11.331418 IP 203.21.20.200.80 > 192.168.1.177.56433: . 323:1771(1448) ack 164 win
1716 <nop,nop,timestamp 3951407263 4452516>
| 13:15:11.331466 IP 192.168.1.177.56433 > 203.21.20.200.80: . ack 1 win 365
<nop,nop,timestamp 4452603 3951407263,nop,nop,sack 1 {323:1771}>
* 13:15:11.332032 IP 203.21.20.200.80 > 192.168.1.177.56433: P 1:323(322) ack 164 win 1716
<nop,nop,timestamp 3951407263 4452516>
| 13:15:11.661515 IP 203.21.20.200.80 > 192.168.1.177.56433: . 1771:3219(1448) ack 164 win
1716 <nop,nop,timestamp 3951407594 4452603>
| 13:15:12.701158 IP 203.21.20.200.80 > 192.168.1.177.56433: P 1:323(322) ack 164 win 1716
<nop,nop,timestamp 3951408634 4452603>
| 13:15:15.443709 IP 203.21.20.200.80 > 192.168.1.177.56433: P 1:323(322) ack 164 win 1716
<nop,nop,timestamp 3951411376 4452603>
| 13:15:20.925462 IP 203.21.20.200.80 > 192.168.1.177.56433: P 1:323(322) ack 164 win 1716
<nop,nop,timestamp 3951416860 4452603>
| 13:15:31.893838 IP 203.21.20.200.80 > 192.168.1.177.56433: P 1:323(322) ack 164 win 1716
<nop,nop,timestamp 3951427828 4452603>
| 13:15:53.823136 IP 203.21.20.200.80 > 192.168.1.177.56433: P 1:323(322) ack 164 win 1716
<nop,nop,timestamp 3951449764 4452603>
| 13:16:37.687442 IP 203.21.20.200.80 > 192.168.1.177.56433: P 1:323(322) ack 164 win 1716
<nop,nop,timestamp 3951493636 4452603>

The tcprcvbuf line for this VE was:

tcprcvbuf	182456	303744	159744	262144	18

The peak usage is higher than the limits, because I did another transfer before
lowering the limits and trying this one.

--
Marcin Owsiany <marcin@owsiany.pl> http://marcin.owsiany.pl/
GnuPG: 1024D/60F41216 FE67 DA2D 0ACA FC5E 3F75 D6F6 3A0D 8AA0 60F4 1216

"Every program in development at MIT expands until it can read mail."
-- Unknown

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by gblond on Wed, 26 Mar 2008 12:13:56 GMT
View Forum Message <> Reply to Message

On 25 March 2008 15:50:46 Marcin Owsiany wrote:
> On Tue, Mar 25, 2008 at 03:33:08PM +0300, Vitaliy Gusev wrote:
> > Does the original kernel 028.53.3 still have issue with sending ack?
>
> I didn't try.
>
> > I see if VE has too small tcprcvbuf (about 30000) then server retransmit
> > packets to VE. But i can't reproduce a deadlock state.
>
> Maybe you need a crappy internet connection to be able to reproduce this
> :)
>
> The way I did that recently was:
> 1) saturate the link (medium-quality 4 Mb ADSL) in both directions from
> another machine

Traffic from other machine to VE (and vise-versa) or from other machine to VE0?

Is it TCP or ICMP traffic?

>
> 2) give it a couple of minutes until the transfer rates are steady
>
> 3) start downloading a large file in a VE (I'm in UK and used a
> debian-cd mirror in Australia for that)
>
> 4) give it a couple of minutes until the transfer rate is steady (you
> can observe the current window size in tcpdump - just wait until it
> stops changing)
>
> 5) then stop both transfers on the other machine
>
> 6) after some time (10-20 seconds), the testing VE will notice that
> more bandwidth has become available, and you will notice that the
> window size will start increasing, to increase the transfer rate

>
> 7) for me, at the point the window size has reached about 9KB (it takes
> just a couple of seconds from the moment the window started
> increasing), the deadlock occured
>

--
Thank,
Vitaliy Gusev

Subject: Re: Problem with TCP window too large for TCPRCVBUF still present
Posted by porridge on Wed, 26 Mar 2008 12:31:20 GMT
View Forum Message <> Reply to Message

On Wed, Mar 26, 2008 at 03:13:56PM +0300, Vitaliy Gusev wrote:
> > The way I did that recently was:
> > 1) saturate the link (medium-quality 4 Mb ADSL) in both directions from
> > another machine
>
> Traffic from other machine to VE (and vise-versa) or from other machine to VE0?

```
 [ Box A ]--\
            +----<ADSL>-----[ Internet ]------[ cd mirror in australia ]
 [ Box B ]--/
```

Box A contains the VE I'm testing the patch in.
Box B is another box on the same LAN, using the same ADSL line, which I
used to generate the traffic on the internet link in point "1)".

> Is it TCP or ICMP traffic?

TCP (HTTP download and an scp upload).

--
Marcin Owsiany <marcin@owsiany.pl> http://marcin.owsiany.pl/
GnuPG: 1024D/60F41216 FE67 DA2D 0ACA FC5E 3F75 D6F6 3A0D 8AA0 60F4 1216

"Every program in development at MIT expands until it can read mail."
-- Unknown
