Subject: [patch 0/3][netns] fix and wipeout timewait sockets
Posted by Daniel Lezcano on Mon, 24 Sep 2007 13:29:35 GMT
View Forum Message <> Reply to Message

Denis Lunev spotted that using a reference to the network namespace
with the timewait sockets will be a waste of time because they
are pointless while we will remove the network stack at network
namespace exit.

The following patches do the following:
 - fix missing network namespace reference in timewait
 socket
 - do some changes in timewait socket code to prepare
 the next patch, especially split code taking a lock
 - do the effective timewait socket cleanup at network
 namespace exit.

The following code is a test program which creates 100 timewait
sockets.

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/poll.h>
#include <netinet/in.h>
#include <netinet/tcp.h>
#include <arpa/inet.h>

#include <unistd.h>

#define MAXCONN 100

int client(int *fds)
{
	int i;
	struct sockaddr_in addr;

	close(fds[1]);

	memset(&addr, 0, sizeof(addr));

	addr.sin_family = AF_INET;
	addr.sin_port =  htons(10000);
	addr.sin_addr.s_addr = inet_addr("127.0.0.1");

	if (read(fds[0], &i, sizeof(i)) == -1) {
		perror("read");
```

```
		return 1;
	}

	for (i = 0; i < MAXCONN; i++) {
		int fd = socket(PF_INET, SOCK_STREAM, 0);
		if (fd == -1) {
			perror("socket");
			return 1;
		}

		if (connect(fd, (const struct sockaddr *)&addr, sizeof(addr))) {
			perror("connect");
			return 1;
		}
	}

	return 0;
}

int server(int *fds)
{
	int i, fd, fdpoll[MAXCONN];
	struct sockaddr_in addr;
	socklen_t socklen = sizeof(addr);

	close(fds[0]);

	fd = socket(PF_INET, SOCK_STREAM, 0);
	if (fd == -1) {
		perror("socket");
		return 1;
	}

	memset(&addr, 0, sizeof(addr));

	addr.sin_family = AF_INET;
	addr.sin_port = htons(10000);
	addr.sin_addr.s_addr = inet_addr("127.0.0.1");

	if (bind(fd, (const struct sockaddr *)&addr, sizeof(addr))) {
		perror("bind");
		return 1;
	}

	if (listen(fd, MAXCONN)) {
		perror("listen");
		return 1;
	}
```

```
	if (write(fds[1], &i, sizeof(i)) == -1) {
		perror("write");
		return 1;
	}

	for (i = 0; i < MAXCONN; i++) {
		int f = accept(fd, (struct sockaddr *)&addr, &socklen);
		if (f == -1) {
			perror("accept");
			return 1;
		}
		fdpoll[i] = f;
	}

	return 0;
}

int main(int argc, char *argv[])
{
	int fds[2];
	int pid;

	if (pipe(fds)) {
		perror("pipe");
		return 1;
	}

	pid = fork();
	if (pid == -1) {
		perror("fork");
		return 1;
	}

	if (!pid)
		return client(fds);
	else
		return server(fds);
}
```

--

_____

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

Subject: [patch 3/3][netns] remove timewait sockets at cleanup
Posted by Daniel Lezcano on Mon, 24 Sep 2007 13:29:38 GMT
View Forum Message <> Reply to Message

From: Daniel Lezcano <dlezcano@fr.ibm.com>

Denis Lunev spotted that if we take a reference to the network namespace
with the timewait sockets, we will need to wait for their expiration to
have the network namespace freed. This is a waste of time, the timewait
sockets are for avoiding to receive a duplicate packet from the network,
if the network namespace is freed, the network stack is removed, so no
chance to receive any packets from the outside world.

This patchset remove/destroy the timewait sockets when the
network namespace is freed.

Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>
---
 net/core/net_namespace.c | 55 ++
 1 file changed, 55 insertions(+)

Index: linux-2.6-netns/net/core/net_namespace.c
===
--- linux-2.6-netns.orig/net/core/net_namespace.c
+++ linux-2.6-netns/net/core/net_namespace.c
@@ -7,6 +7,7 @@
 #include <linux/sched.h>
 #include <linux/kallsyms.h>
 #include <net/net_namespace.h>
+#include <net/tcp.h>

 /*
  * Our network namespace constructor/destructor lists
@@ -59,6 +60,57 @@ static int pernet_debug_max=-1;
 module_param(pernet_debug, int, S_IRUSR|S_IWUSR);
 module_param(pernet_debug_max, int, S_IRUSR|S_IWUSR);

+/*
+ * This function is called when the network namespace is freed.
+ * It allows to wipe out all timewait sockets. They are useless
+ * because the network namespace will be destroyed and the network
+ * stack with it, so no risks to have a duplicate packet coming
+ * from the outside world.
+ */
+static void clean_up_timewait(struct net *net)
+{
+ struct inet_timewait_sock *tw;
+ struct sock *sk;
+ struct hlist_node *node, *tmp;

```
+ int h;
+
+ /* Browse the the established hash table */
+ for (h = 0; h < (tcp_hashinfo.ehash_size); h++) {
+               struct inet_ehash_bucket *head =
+                       inet_ehash_bucket(&tcp_hashinfo, h);
+
+  /* Take the look and disable bh */
+   write_lock_bh(&head->lock);
+
+  sk_for_each_safe(sk, node, tmp, &head->twchain) {
+
+   tw = inet_twsk(sk);
+   if (tw->tw_net != net)
+    continue;
+
+   /* deschedule the timewait socket */
+   spin_lock(&tcp_death_row.death_lock);
+   if (inet_twsk_del_dead_node(tw)) {
+    inet_twsk_put(tw);
+    if (--tcp_death_row.tw_count == 0)
+     del_timer(&tcp_death_row.tw_timer);
+   }
+   spin_unlock(&tcp_death_row.death_lock);
+
+   /* remove it from the established hash table */
+   __inet_twsk_unehash(tw);
+
+   /* remove it from the bind hash table */
+   inet_twsk_unbhash(tw, tcp_death_row.hashinfo);
+
+   /* last put */
+   inet_twsk_put(tw);
+  }
+
+  write_unlock_bh(&head->lock);
+ }
+}
+
 static void cleanup_net(struct work_struct *work)
 {
  struct pernet_operations *ops;
@@ -96,6 +148,9 @@ static void cleanup_net(struct work_stru

  mutex_unlock(&net_mutex);

+ /* The timewait sockets are pointless */
+ clean_up_timewait(net);
```

```
+
  /* Ensure there are no outstanding rcu callbacks using this
   * network namespace.
   */

--
```

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [patch 3/3][netns] remove timewait sockets at cleanup
Posted by ebiederm on Wed, 26 Sep 2007 19:22:26 GMT
View Forum Message <> Reply to Message

Daniel Lezcano <dlezcano@fr.ibm.com> writes:

> From: Daniel Lezcano <dlezcano@fr.ibm.com>
>
> Denis Lunev spotted that if we take a reference to the network namespace
> with the timewait sockets, we will need to wait for their expiration to
> have the network namespace freed. This is a waste of time, the timewait
> sockets are for avoiding to receive a duplicate packet from the network,
> if the network namespace is freed, the network stack is removed, so no
> chance to receive any packets from the outside world.
>
> This patchset remove/destroy the timewait sockets when the
> network namespace is freed.

This code is in the wrong place. Please do the register_net_subsys
thing so we can keep the code in net/ipv4/inet_timewait_sock.c

This code just need to be an exit method.

Eric

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [patch 0/3][netns] fix and wipeout timewait sockets
Posted by ebiederm on Wed, 26 Sep 2007 19:24:14 GMT
View Forum Message <> Reply to Message

Daniel Lezcano <dlezcano@fr.ibm.com> writes:

> Denis Lunev spotted that using a reference to the network namespace
> with the timewait sockets will be a waste of time because they
> are pointless while we will remove the network stack at network
> namespace exit.

The patches look close and look like they are in the right general
direction.

I haven't reviewed them closely yet, so I suspect there are
a few more nits but generally they look good.

Please don't let me forget about this issue.

Thanks,
Eric

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

Subject: Re: [patch 3/3][netns] remove timewait sockets at cleanup
Posted by Daniel Lezcano on Thu, 27 Sep 2007 08:36:17 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> Daniel Lezcano <dlezcano@fr.ibm.com> writes:
>
>> From: Daniel Lezcano <dlezcano@fr.ibm.com>
>>
>> Denis Lunev spotted that if we take a reference to the network namespace
>> with the timewait sockets, we will need to wait for their expiration to
>> have the network namespace freed. This is a waste of time, the timewait
>> sockets are for avoiding to receive a duplicate packet from the network,
>> if the network namespace is freed, the network stack is removed, so no
>> chance to receive any packets from the outside world.
>>
>> This patchset remove/destroy the timewait sockets when the
>> network namespace is freed.
>
> This code is in the wrong place. Please do the register_net_subsys
> thing so we can keep the code in net/ipv4/inet_timewait_sock.c
>
> This code just need to be an exit method.

Thanks Eric, I will fix that.

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers