Subject: *SOLVED* too many of orphaned sockets
Posted by hvdkamer on Tue, 29 Aug 2006 17:46:36 GMT
View Forum Message <> Reply to Message

I've created a setup where on one VE Lighttpd is a name-based proxy and is redirecting to anaother VE with an internal IP-address. That works. So I wanted to test how fast it is and then I ran into problems with the following ab2:

hoefnix:~# ab2 -c 12 -n 2000 http://ve108.armorica.tk/

Below a concurrency of 8 everything is fine, between 9 and 11 it sometimes goes well. From 12 and upwards is goes always wrong with some failed requests. On the hardware node I then get the following message:

Aug 29 18:23:08 strato kernel: printk: 2 messages suppressed.
Aug 29 18:23:08 strato kernel: TCP: too many of orphaned sockets
Aug 29 18:23:08 strato last message repeated 9 times

This is bullshit however . The tcp_max_orphans is 32.768. With an constant cat /proc/net/sockstat I see that the orphans are not raised. However because of the setup I do see 4.000 time_wait buckets which die after two minutes. The user_beancounters in bothe VE's are still zero, even after multiple runs.

I'm not an programmer, but just to see when this message is given leads to tcp.c with the following code:

```
if (sk->sk_state != TCP_CLOSE) {
        sk_stream_mem_reclaim(sk);
        if (tcp_too_many_orphans(sk, tcp_get_orphan_count(sk))) {
                if (net_ratelimit())
                        printk(KERN_INFO "TCP: too many of orphaned "
                               "sockets\n");
                tcp_set_state(sk, TCP_CLOSE);
                tcp_send_active_reset(sk, GFP_ATOMIC);
                NET_INC_STATS_BH(LINUX_MIB_TCPABORTONMEMORY);
        }
}
```

And the function tcp_too_may_orphans leads to a file ub_orphan.h which is copyrighted by SWsoft. So I think I'm her at the right source . Can someone give an clue for which parameter I must tune? It isn't one of the beancounters (all zero) or tcp_max_orphans (never reached). There are some other things checked in this function, but taht is way above my head. Pleas advice...

Subject: Re: too many of orphaned sockets
Posted by Vasily Tarasov on Wed, 30 Aug 2006 05:50:23 GMT
View Forum Message <> Reply to Message

So you're using 2.6.16 series...
Look at the code:

```
static inline int ub_too_many_orphans(struct sock *sk, int count)
{
#ifdef CONFIG_USER_RESOURCE
	if (__ub_too_many_orphans(sk, count))			# MAY BE WE HAVE 1 HERE?
		return 1;
#endif
	return (ub_get_orphan_count(sk) > sysctl_tcp_max_orphans ||
		(sk->sk_wmem_queued > SOCK_MIN_SNDBUF &&
		 atomic_read(&tcp_memory_allocated) > sysctl_tcp_mem[2]));
}
```

So, what we have in __ub_too_many_orphans(sk, count):

```
int __ub_too_many_orphans(struct sock *sk, int count)
{
	struct user_beancounter *ub;

	if (sock_has_ubc(sk)) {
		for (ub = sock_bc(sk)->ub; ub->parent != NULL; ub = ub->parent);
		if (count >= ub->ub_parms[UB_NUMTCPSOCK].barrier >> 2)		# IT HOLDS
TRUE
			return 1;
	}
	return 0;
}
```

So the number of orphaned sockets (count) is greater, then (barrier of NUMTCPSOCK parameter) /4. Thus, if the reason is that, you can increase the barrier (not limit!) of numtcpsock parameter.

HTH.

Subject: Re: too many of orphaned sockets
Posted by hvdkamer on Wed, 30 Aug 2006 07:48:50 GMT
View Forum Message <> Reply to Message

vass wrote on Wed, 30 August 2006 07:50So you're using 2.6.16 series...

Nope, the 2.6.8 series . But I think the functions are the same.

vass wrote on Wed, 30 August 2006 07:50Look at the code:

As said, I'm not a C programmer . But if I understand you correctly, the second return with the sysctl_tcp_max_orphans is never reached. So indeed this function is replaced with a different accounting for a VE? Ok, that will explain that my experimenting with the parameters didn't solve anything .

vass wrote on Wed, 30 August 2006 07:50So the number of orphaned sockets (count) is greater, then (barrier of NUMTCPSOCK parameter) /4. Thus, if the reason is that, you can increase the barrier (not limit!) of numtcpsock parameter.

But is it possible to use a higher barrier than the limit? Because the limit is never reached, the failcnt is still zero. Anyway, I wil experiment with this parameter to see if it will surpress the message and if I get better results with the Apache Benchmark. Let you know.

Subject: Re: too many of orphaned sockets
Posted by Vasily Tarasov on Wed, 30 Aug 2006 08:33:02 GMT
View Forum Message <> Reply to Message

Hmmm... And what particular kernel version do you use?..
I'm asking you, 'cause kernel code you've posted in your _first_ post is in 2.6.16 series (at list in 2.6.16-026test017.1). And in 2.6.8-022stab078.14 it differs:

```
if (sk->sk_state != TCP_CLOSE) {
        sk_stream_mem_reclaim(sk);
        if (atomic_read(&tcp_orphan_count) > sysctl_tcp_max_orphans ||
            (sk->sk_wmem_queued > SOCK_MIN_SNDBUF &&
             atomic_read(&tcp_memory_allocated) > sysctl_tcp_mem[2])) {
                if (net_ratelimit())
                        printk(KERN_INFO "TCP: too many of orphaned "
                               "sockets\n");
                tcp_set_state(sk, TCP_CLOSE);
                tcp_send_active_reset(sk, GFP_ATOMIC);
                NET_INC_STATS_BH(LINUX_MIB_TCPABORTONMEMORY);
        }
}
```

Subject: Re: too many of orphaned sockets
Posted by Vasily Tarasov on Wed, 30 Aug 2006 08:40:05 GMT
View Forum Message <> Reply to Message

Sorry... my fault. I used wrong kernel. However precise kernel version is required.

Subject: Re: too many of orphaned sockets
Posted by hvdkamer on Wed, 30 Aug 2006 13:47:29 GMT
View Forum Message <> Reply to Message

Well, despite which kernel, I think your explanation is stille the right one. Because now I know that it is a 1/4 of the barier, I did manage to sqeeuze the maximum out of a very minimal VE .

First parameter I forgot is tuning the TCP sockets of the proxy Lighttpd server. It uses two for every request. One from him to the visitor and one to the correct, internal miniserver. That one could go to 32 simultanous connections, so I scaled the first to 64 (it was 48).

My next assumption was that probably every concurrent connection in the Apache benchmark could give an orpahed connection. That explains probably why with -c 10 it goes most of the time alriight and with 12 the 1/4 of 48 is reached. And indeed I found out that every increase in the -c parameter must raise the barrier of the proxy VE. With that I could go as high as -c 28 (because that raised the maxheld to 31) if I set it to 112:64. That one is illegal according to vzcfgvalidate, but you can still set it.

So the only question remaining is why 1/4? They above experiment suggests 1/2. Anyway, I now know what the warning is about and that it is nothing more than that. Thanks for the explanation. It would be great if more of this knowledge is summearized somewehre in the wiki. I saw something about memory, but not this kind of stuff. May be I must start the page myself .