Subject: Re: [PATCH 0/3] Sysctl shadow management
Posted by ebiederm on Tue, 20 Nov 2007 13:05:58 GMT
View Forum Message <> Reply to Message

Pavel Emelyanov <xemul@openvz.org> writes:

> Hi guys!
>
> You all know, that with multiple namespaces we have to take
> special care about sysctls. E.g. IPC sysctl handlers are
> equipped with kludges to alter the sysctl parameters of
> appropriate namespace. The same thing should be done for UTS
> namespace (but it is not - we have a BUG in mainstream) and
> (!) for network namespaces.

The bug in mainstream was introduced by commit:
7d69a1f4a72b18876c99c697692b78339d491568

Thee code should read:

```
 static void *get_uts(ctl_table *table, int write)
 {
  char *which = table->data;
+ struct uts_namespace *uts_ns = current->nsproxy->uts_ns;
+ which = (which - (char *)&init_uts_ns) + (char *)uts_ns;

  if (!write)
   down_read(&uts_sem);
  else
   down_write(&uts_sem);
  return which;
 }
```

And for 2.6.24 we should just restore the two missing lines.

> Unlike all the other namespaces, network will have to not
> just address different variables via same sysctl names, but
> to have different tables with different sysctl names. E.g.
> /proc/sys/net/conf have entries for devices, which differ
> across namespaces.
>
> Eric currently have some work done in that directions, I
> like the approach in general very much, but it looks rather
> raw (Eric, take this in good part). You know, ifdefs in the
> middle of the code, explicit references to net namespace
> and so on and so forth.

Sure. I don't in principle have any problem with the set of roots we go through be dynamic at run time instead of compile time. That is probably a cleaner approach and likely to solve my problem with sched_debug registering sysctls before the sysctl subsystem is initialized.

One direction I have always intended to expand things when there was a bit of time to modify /proc/sys so that it is a symlink to /proc/self/sys. Then make /proc/<pid>/sys have cachable dentries for the sysctls.

To achieve that we need to pass our namespaces into your shadow function. Instead of always using current.

So I am thinking something like:

```
struct ctl_table_root {
 struct list_head ctl_entry;
 struct ctl_table_header *ctl_head;
 struct ctl_table_header *lookup_ctl_head(struct nsproxy *namespaces);
};
```

To actually handle the set of network devices in a namespace we need to have a list so making sysctl_head_next just loop over a list of lists should be no extra work and make implementing the users easier.

Beyond that through from my quick skim I have a preference for the way I am handling it.

We really need to add the tables with some variant of register_sysctl_table or else we will have module unload races.

Introducing register_sysctl_paths is a very useful cleanup in it's own right and it helps quite a bit. In most cases it removes the need for your create_sysctl_shadow function, and it always reduced the amount of code for tables.

Further simply using kmemdup instead of a custom crafted function is a little more straight forward and is the idiom already established throughout the networking code.

Then we will just need a base sysctl function:

```
struct ctl_table_header *
register_sysctl_rooted_paths(struct ctl_table_root *root,
    struct nsproxy *namespaces,
    struct ctl_path *path,
    struct ctl_table *table)
```

For the simple namespaces we can call it once per namespace
after registering our root (we can't use current because we initialize
things before we update nsproxy), and we still need to run
sysctl_check.

For the more complex namespaces (i.e. the network namespace). We can
write a simple wrapper around register_sysctl_rooted_paths.

And of course non-namespace specific sysctls can just use a wrapper
that assumes the default global list of sysctl_headers.

> So here's the RFC for a bit better sysctls shadow management.
>
> I will provide 3 patches:
> 1. the sysctl shadows themselves;
> 2. using shadows in UTS namespace;
> 3. using shadows in IPC namespace;

Your patches look fairly reasonable. Other then the pieces
I have mentioned already.

> Using them in net namespace is already checked (I created
> sysctl entries with different names), but I don't have any
> patches against any David's tree yet. If we're OK with this
> set I will start talking to Andrew and David about who to
> send these patches to and making shadows for net-related
> sysctl variables.

I think we need another round. My hunch is that it will be easiest
if David collects them up, and then Andrew updates his tree, but
we will see.

Eric