Subject: Re: [PATCH 0/4] Devices accessibility control group (v2)
Posted by Sukadev Bhattiprolu on Sat, 12 Jan 2008 21:20:14 GMT
View Forum Message <> Reply to Message

```
Pavel Emelianov [xemul@openvz.org] wrote:
| The first version was posted long ago
| (http://openvz.org/pipermail/devel/2007-September/007647.html)
| and since then there are many (good I hope) changes:
|
| * Added the block devices support :) It turned out to
|   be a bit simpler than the char one (or I missed
|   something significant);
| * Now we can enable/disable not just individual devices,
|   but the whole major with all its minors (see the TODO
|   list beyond as well);
| * Added the ability to restrict the read/write permissions
|   to devices, not just visible/invisible state.
|
| That is - the main features I wished to implement right
| after the v1 was sent. Some minor changes are:
|
| * I merged the devices.char and devices.block files into
|   one - devices.permissions;
| * As the result of the change above - the strings passed
|   to this file has changed. Now they are
|         [bc] <major>:{<minor>|*} [r-][w-]
|   E.g. b 5:2 r- will grant the read permissions to the
|   block 5:2 device and c 3:* -w will grant the write-only
|   access to all the character devices with the major 5.
|
| However, there are some things to be done:
|
| * Make the /proc/devices show relevant info depending on
|   who is reading it. This seems to be easy to do, since
|   I already have the support to dump similar info into the
|   devices.permissions file, but I haven't tried to use
|   this in /proc/devices yet;
| * Add the support for devices ranges. I.e. someone might
|   wish to tell smth like b 5:[0-10] r- to this subsystem.
|   Currently this is not supported and I'm afraid that if we
|   start support minor ranges we'll have smth similar to
|   VMA-s or FLOCK-s ranges management in one more place in the
|   kernel.
| * One more question is - are there any other permissions to
|   work with? E.g. in OpenVZ we have a separate bit for
|   quota management, maybe we can invent some more...
|
| Currently I didn't pay much attention to split this set well,
```

| so this will most likely won't work with git-bisect, but I
| think this is OK for now. I will sure split it better when I
| send the v3 and further.
|
| The set is prepared against the 2.6.24-rc5-mm1.
|
| All this is minimally tested and seems to work. Hope to hear
| you comments, wishes and patches soon :)
|
| To play with it - run a standard procedure:
|
| # mount -t container none /cont/devs -o devices

This should be '-t cgroup'

| # mkdir /cont/devs/0
| # echo -n $$ > /cont/devs/0/tasks
|
| and tune device permissions.

I started playing with this and noticed that even if I try to
enable read access to device [c, 1:3] it also grants access
to device [c, 1:5].

i.e the access restrictions seem to apply to all devices with
a given major number. Is that really the intent ?

Both devices accessible here:

```
 # hexdump /dev/null
 # hexdump /dev/zero
 0000000 0000 0000 0000 0000 0000 0000 0000 0000
 *
 ^C
```

Neither device accessible:

```
 # echo $$ > /container/devs/0/tasks
 # hexdump /dev/zero
 hexdump: /dev/zero: No such device or address
 hexdump: /dev/zero: Bad file descriptor
 # hexdump /dev/null
 hexdump: /dev/null: No such device or address
 hexdump: /dev/null: Bad file descriptor
```

Grant read access to /dev/null, but /dev/zero is also readable

```
 # echo c 1:3 r- > /container/devs/0/devices.permissions
 # hexdump /dev/null
 # hexdump /dev/zero
 0000000 0000 0000 0000 0000 0000 0000 0000 0000
 *
 ^C
```

Remove read access to /dev/null, but /dev/zero is also not
readable.

```
 # echo c 1:3 -- > /container/devs/0/devices.permissions
 # hexdump /dev/zero
 hexdump: /dev/zero: No such device or address
 hexdump: /dev/zero: Bad file descriptor
```

BTW, a question about cgroups: If we 'echo $$ > /container/devs/0/tasks'
is there a way to remove/undo it later (so that the process has access
as before) ?

____