
Subject: Re: [patch 1/2] [RFC] Simple tamper-proof device filesystem.

Posted by [serue](#) on Mon, 17 Dec 2007 19:48:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quoting Tetsuo Handa (penguin-kernel@I-love.SAKURA.ne.jp):

> A brief description about SYAORAN:

>

> SYAORAN stands for "Simple Yet All-important Object Realizing Abiding
> Nexus". SYAORAN is a filesystem for /dev with Mandatory Access Control.

>

> /dev needs to be writable, but this means that files on /dev might be
> tampered with. SYAORAN can restrict combinations of (pathname, attribute)
> that the system can create. The attribute is one of directory, regular
> file, FIFO, UNIX domain socket, symbolic link, character or block device
> file with major/minor device numbers.

>

> SYAORAN can ensure /dev/null is a character device file with major=1 minor=3.

>

> Policy specifications for this filesystem is at
> <http://tomoyo.sourceforge.jp/en/1.5.x/policy-syaoran.html>

>

> Why not use FUSE?

>

> Because /dev has to be available through the lifetime of the kernel.
> It is not acceptable if /dev stops working due to SIGKILL or OOM-killer.

>

> Why not use SELinux?

>

> Because SELinux doesn't guarantee filename and its attribute.
> The purpose of this filesystem is to ensure filename and its attribute
> (e.g. /dev/null is guaranteed to be a character device file
> with major=1 and minor=3).

We need something similar for system containers (like vservers). We will likely want root in a container to be confined to a certain set of devices.

For starters we expect to use the capability bounding sets (see <http://lkml.org/lkml/2007/11/26/206>). So a container will have a static /dev predefined, and CAP_MKNOD will be removed from its capability bounding set so that root in a container cannot create any more new devices.

For future more sophisticated device controls, two similar approaches have been suggested (one by me, see <https://lists.linux-foundation.org/pipermail/containers/2007-September/007423.html> and <https://lists.linux-foundation.org/pipermail/containers/2007-November/008589.html>

). Both actually control the devices a process can create period, rather than trying to control at the filesystem. And yes, these both lack the feature in your solution that for instance 'c 1 3' must be called null, which appears to be the kind of guarantee apparmor likes to provide.

To use your approach, i guess we would have to use selinux (or tomoyo) to enforce that devices may only be created under /dev?

-serge

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

Subject: Re: [patch 1/2] [RFC] Simple tamper-proof device filesystem.
Posted by [Tetsuo Handa](#) on Tue, 18 Dec 2007 00:03:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello.

Serge E. Hallyn wrote:

> CAP_MKNOD will be removed from its capability

I think it is not enough because the root can rename/unlink device files
(mv /dev/sda1 /dev/tmp; mv /dev/sda2 /dev/sda1; mv /dev/tmp /dev/sda2).

> To use your approach, i guess we would have to use selinux (or tomoyo)

> to enforce that devices may only be created under /dev?

Everyone can use this filesystem alone.

But use with MAC (or whatever access control mechanisms that prevent attackers from unmounting/overlaying this filesystem) is recommended.

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