Subject: Re: [patch 1/2] [RFC] Simple tamper-proof device filesystem.
Posted by serue on Tue, 18 Dec 2007 00:39:55 GMT
View Forum Message <> Reply to Message

Quoting Tetsuo Handa (penguin-kernel@i-love.sakura.ne.jp):
> 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).

Sure but that doesn't bother us :)

The admin in the container has his own /dev directory and can do what he
likes with the devices he's allowed to have. He just shouldn't have
access to others. If he wants to rename /dev/sda1 to /dev/sda5 that's
his choice.

> > 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.

Sure but it is worthless alone.

No?

What will keep the container admin from doing 'mknod /root/hda1 b 3 1'?

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

-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 Oren Laadan on Tue, 18 Dec 2007 01:39:47 GMT
View Forum Message <> Reply to Message

I hate to bring this again, but what if the admin in the container
mounts an external file system (eg. nfs, usb, loop mount from a file,
or via fuse), and that file system already has a device that we would
like to ban inside that container ?

Since anyway we will have to keep a white- (or black-) list of devices
that are permitted in a container, and that list may change even change
per container -- why not enforce the access control at the VFS layer ?
It's safer in the long run.

Oren.

Serge E. Hallyn wrote:
> Quoting Tetsuo Handa (penguin-kernel@i-love.sakura.ne.jp):
>> 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).
>
> Sure but that doesn't bother us :)
>
> The admin in the container has his own /dev directory and can do what he
> likes with the devices he's allowed to have. He just shouldn't have
> access to others. If he wants to rename /dev/sda1 to /dev/sda5 that's
> his choice.
>
>>> 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.
>
> Sure but it is worthless alone.
>
> No?
>
> What will keep the container admin from doing 'mknod /root/hda1 b 3 1'?
>
>> But use with MAC (or whatever access control mechanisms that prevent
>> attackers from unmounting/overlaying this filesystem) is recomennded.
>
> -serge
> _____
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containers

_____
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 serue on Tue, 18 Dec 2007 01:55:57 GMT
View Forum Message <> Reply to Message

Quoting Serge E. Hallyn (serue@us.ibm.com):
> Quoting Tetsuo Handa (penguin-kernel@i-love.sakura.ne.jp):
> > 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).
>
> Sure but that doesn't bother us :)
>
> The admin in the container has his own /dev directory and can do what he
> likes with the devices he's allowed to have. He just shouldn't have
> access to others. If he wants to rename /dev/sda1 to /dev/sda5 that's
> his choice.
>
> > > 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.
>
> Sure but it is worthless alone.
>
> No?

Oh, no, I'm sorry - I was thinking in terms of my requirements again.
But your requirements are to ensure that an application accessing a
device at a well-known location get what it expect.

So then the main quesiton is still the one I think Al had asked - what
keeps a rogue CAP_SYS_MOUNT process from doing
mount --bind /dev/hda1 /dev/null ?

thanks,
-serge

> What will keep the container admin from doing 'mknod /root/hda1 b 3 1'?
>
> > But use with MAC (or whatever access control mechanisms that prevent
> > attackers from unmounting/overlaying this filesystem) is recomennded.
>
> -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 serue on Tue, 18 Dec 2007 02:09:33 GMT
View Forum Message <> Reply to Message

Quoting Oren Laadan (orenl@cs.columbia.edu):
>
> I hate to bring this again, but what if the admin in the container
> mounts an external file system (eg. nfs, usb, loop mount from a file,
> or via fuse), and that file system already has a device that we would
> like to ban inside that container ?

Miklos' user mount patches enforced that if !capable(CAP_MKNOD),
then mnt->mnt_flags |= MNT_NODEV. So that's no problem.

But that's been pulled out of -mm! ? Crap.

> Since anyway we will have to keep a white- (or black-) list of devices
> that are permitted in a container, and that list may change even change
> per container -- why not enforce the access control at the VFS layer ?
> It's safer in the long run.

By that you mean more along the lines of Pavel's patch than my whitelist
LSM, or you actually mean Tetsuo's filesystem (i assume you don't mean that
by 'vfs layer' :), or something different entirely?

thanks,
-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 02:26:21 GMT
View Forum Message <> Reply to Message

Hello.

Serge E. Hallyn wrote:
> But your requirements are to ensure that an application accessing a
> device at a well-known location get what it expect.

Yes. That's the purpose of this filesystem.

> So then the main quesiton is still the one I think Al had asked - what
> keeps a rogue CAP_SYS_MOUNT process from doing

> mount --bind /dev/hda1 /dev/null ?

Excuse me, but I guess you meant "mount --bind /dev/ /root/" or something
because mount operation requires directories.
MAC can prevent a rogue CAP_SYS_MOUNT process from doing
"mount --bind /dev/ /root/".
For example, regarding TOMOYO Linux, you need to give
"allow_mount /dev/ /root/ --bind 0" permission
to permit "mount --bind /dev/ /root/" request.

Did you mean "ln -s /dev/hda1 /dev/null" or "ln /dev/hda1 /dev/null"?
No problem. MAC can prevent such requests too.

Regards.

Subject: Re: [patch 1/2] [RFC] Simple tamper-proof device filesystem.
Posted by Oren Laadan on Tue, 18 Dec 2007 03:03:42 GMT
View Forum Message <> Reply to Message

Serge E. Hallyn wrote:
> Quoting Oren Laadan (orenl@cs.columbia.edu):
>> I hate to bring this again, but what if the admin in the container
>> mounts an external file system (eg. nfs, usb, loop mount from a file,
>> or via fuse), and that file system already has a device that we would
>> like to ban inside that container ?
>
> Miklos' user mount patches enforced that if !capable(CAP_MKNOD),
> then mnt->mnt_flags |= MNT_NODEV. So that's no problem.

Yes, that works to disallow all device files from a mounted file system.

But it's a black and white thing: either they are all banned or allowed;
you can't have some devices allowed and others not, depending on type
A scenario where this may be useful is, for instance, if we some apps in
the container to execute withing a pre-made chroot (sub)tree within that
container.

>
> But that's been pulled out of -mm! ? Crap.
>
>> Since anyway we will have to keep a white- (or black-) list of devices
>> that are permitted in a container, and that list may change even change
>> per container -- why not enforce the access control at the VFS layer ?

>> It's safer in the long run.
>
> By that you mean more along the lines of Pavel's patch than my whitelist
> LSM, or you actually mean Tetsuo's filesystem (i assume you don't mean that
> by 'vfs layer' :), or something different entirely?

:)

By 'vfs' I mean at open() time, and not at mount(), or mknod() time.
Either yours or Pavel's; I tend to prefer not to use LSM as it may
collide with future security modules.

Oren.

>
> thanks,
> -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 Pavel Emelianov on Wed, 19 Dec 2007 09:43:14 GMT
View Forum Message <> Reply to Message

Oren Laadan wrote:
> Serge E. Hallyn wrote:
>> Quoting Oren Laadan (orenl@cs.columbia.edu):
>>> I hate to bring this again, but what if the admin in the container
>>> mounts an external file system (eg. nfs, usb, loop mount from a file,
>>> or via fuse), and that file system already has a device that we would
>>> like to ban inside that container ?
>> Miklos' user mount patches enforced that if !capable(CAP_MKNOD),
>> then mnt->mnt_flags |= MNT_NODEV. So that's no problem.
>
> Yes, that works to disallow all device files from a mounted file system.
>
> But it's a black and white thing: either they are all banned or allowed;
> you can't have some devices allowed and others not, depending on type
> A scenario where this may be useful is, for instance, if we some apps in
> the container to execute withing a pre-made chroot (sub)tree within that
> container.
>
>> But that's been pulled out of -mm! ? Crap.
>>
>>> Since anyway we will have to keep a white- (or black-) list of devices

>>> that are permitted in a container, and that list may change even change
>>> per container -- why not enforce the access control at the VFS layer ?
>>> It's safer in the long run.
>> By that you mean more along the lines of Pavel's patch than my whitelist
>> LSM, or you actually mean Tetsuo's filesystem (i assume you don't mean that
>> by 'vfs layer' :), or something different entirely?
>
> :)
>
> By 'vfs' I mean at open() time, and not at mount(), or mknod() time.
> Either yours or Pavel's; I tend to prefer not to use LSM as it may
> collide with future security modules.

Oren, AFAIS you've seen my patches for device access controller, right?

Maybe we can revisit the issue then and try to come to agreement on what
kind of model and implementation we all want?

> Oren.
>
>> thanks,
>> -serge
> --
> To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
> the body of a message to majordomo@vger.kernel.org
> More majordomo info at http://vger.kernel.org/majordomo-info.html
> Please read the FAQ at http://www.tux.org/lkml/
>

Subject: Re: [patch 1/2] [RFC] Simple tamper-proof device filesystem.
Posted by serue on Wed, 19 Dec 2007 14:10:04 GMT
View Forum Message <> Reply to Message

Quoting Pavel Emelyanov (xemul@openvz.org):
> Oren Laadan wrote:
> > Serge E. Hallyn wrote:
> >> Quoting Oren Laadan (orenl@cs.columbia.edu):
> >>> I hate to bring this again, but what if the admin in the container
> >>> mounts an external file system (eg. nfs, usb, loop mount from a file,
> >>> or via fuse), and that file system already has a device that we would
> >>> like to ban inside that container ?
> >> Miklos' user mount patches enforced that if !capable(CAP_MKNOD),

> >> then mnt->mnt_flags |= MNT_NODEV. So that's no problem.
> >
> > Yes, that works to disallow all device files from a mounted file system.
> >
> > But it's a black and white thing: either they are all banned or allowed;
> > you can't have some devices allowed and others not, depending on type
> > A scenario where this may be useful is, for instance, if we some apps in
> > the container to execute withing a pre-made chroot (sub)tree within that
> > container.
> >
> >> But that's been pulled out of -mm! ? Crap.
> >>
> >>> Since anyway we will have to keep a white- (or black-) list of devices
> >>> that are permitted in a container, and that list may change even change
> >>> per container -- why not enforce the access control at the VFS layer ?
> >>> It's safer in the long run.
> >> By that you mean more along the lines of Pavel's patch than my whitelist
> >> LSM, or you actually mean Tetsuo's filesystem (i assume you don't mean that
> >> by 'vfs layer' :), or something different entirely?
> >
> > :)
> >
> > By 'vfs' I mean at open() time, and not at mount(), or mknod() time.
> > Either yours or Pavel's; I tend to prefer not to use LSM as it may
> > collide with future security modules.
>
> Oren, AFAIS you've seen my patches for device access controller, right?
>
> Maybe we can revisit the issue then and try to come to agreement on what
> kind of model and implementation we all want?

That would be great, Pavel. I do prefer your solution over my LSM, so
if we can get an elegant block device control right in the vfs code that
would be my preference.

The only thing that makes me keep wanting to go back to an LSM is the
fact that the code defining the whitelist seems out of place in the vfs.
But I guess that's actually separated into a modular cgroup, with the
actual enforcement built in at the vfs. So that's really the best
solution.

-serge

Subject: Re: [patch 1/2] [RFC] Simple tamper-proof device filesystem.
Posted by serue on Wed, 19 Dec 2007 14:13:30 GMT
View Forum Message <> Reply to Message

Quoting Oren Laadan (orenl@cs.columbia.edu):
>
> Serge E. Hallyn wrote:
> > Quoting Oren Laadan (orenl@cs.columbia.edu):
> >> I hate to bring this again, but what if the admin in the container
> >> mounts an external file system (eg. nfs, usb, loop mount from a file,
> >> or via fuse), and that file system already has a device that we would
> >> like to ban inside that container ?
> >
> > Miklos' user mount patches enforced that if !capable(CAP_MKNOD),
> > then mnt->mnt_flags |= MNT_NODEV. So that's no problem.
>
> Yes, that works to disallow all device files from a mounted file system.
>
> But it's a black and white thing: either they are all banned or allowed;
> you can't have some devices allowed and others not, depending on type
> A scenario where this may be useful is, for instance, if we some apps in
> the container to execute withing a pre-made chroot (sub)tree within that
> container.

Yes, it's workable short-term, and we've always said that a more
complete solution would be worked on later, as people have time.

> > But that's been pulled out of -mm! ? Crap.
> >
> >> Since anyway we will have to keep a white- (or black-) list of devices
> >> that are permitted in a container, and that list may change even change
> >> per container -- why not enforce the access control at the VFS layer ?
> >> It's safer in the long run.
> >
> > By that you mean more along the lines of Pavel's patch than my whitelist
> > LSM, or you actually mean Tetsuo's filesystem (i assume you don't mean that
> > by 'vfs layer' :), or something different entirely?
>
> :)
>
> By 'vfs' I mean at open() time, and not at mount(), or mknod() time.
> Either yours or Pavel's; I tend to prefer not to use LSM as it may
> collide with future security modules.

Yeah I keep waffling. The LSM is so simple... but i do prefer Pavel's
patch. Let's keep pursuing that.

-serge

Subject: Re: [patch 1/2] [RFC] Simple tamper-proof device filesystem.
Posted by Oren Laadan on Thu, 20 Dec 2007 00:07:04 GMT
View Forum Message <> Reply to Message

Serge E. Hallyn wrote:
> Quoting Pavel Emelyanov (xemul@openvz.org):
>> Oren Laadan wrote:
>>> Serge E. Hallyn wrote:
>>>> Quoting Oren Laadan (orenl@cs.columbia.edu):
>>>>> I hate to bring this again, but what if the admin in the container
>>>>> mounts an external file system (eg. nfs, usb, loop mount from a file,
>>>>> or via fuse), and that file system already has a device that we would
>>>>> like to ban inside that container ?
>>>> Miklos' user mount patches enforced that if !capable(CAP_MKNOD),
>>>> then mnt->mnt_flags |= MNT_NODEV. So that's no problem.
>>> Yes, that works to disallow all device files from a mounted file system.
>>>
>>> But it's a black and white thing: either they are all banned or allowed;
>>> you can't have some devices allowed and others not, depending on type
>>> A scenario where this may be useful is, for instance, if we some apps in
>>> the container to execute withing a pre-made chroot (sub)tree within that
>>> container.
>>>
>>>> But that's been pulled out of -mm! ? Crap.
>>>>
>>>>> Since anyway we will have to keep a white- (or black-) list of devices
>>>>> that are permitted in a container, and that list may change even change
>>>>> per container -- why not enforce the access control at the VFS layer ?
>>>>> It's safer in the long run.
>>>> By that you mean more along the lines of Pavel's patch than my whitelist
>>>> LSM, or you actually mean Tetsuo's filesystem (i assume you don't mean that
>>>> by 'vfs layer' :), or something different entirely?
>>> :)
>>>
>>> By 'vfs' I mean at open() time, and not at mount(), or mknod() time.
>>> Either yours or Pavel's; I tend to prefer not to use LSM as it may
>>> collide with future security modules.
>> Oren, AFAIS you've seen my patches for device access controller, right?

If you mean this one:
http://openvz.org/pipermail/devel/2007-September/007647.html
then ack :)

>>
>> Maybe we can revisit the issue then and try to come to agreement on what
>> kind of model and implementation we all want?
>
> That would be great, Pavel. I do prefer your solution over my LSM, so
> if we can get an elegant block device control right in the vfs code that
> would be my preference.

I concur.

So it seems to me that we are all in favor of the model where open()
of a device will consult a black/white-list. Also, we are all in favor
of a non-LSM implementation, Pavel's code being a good example.

Oren.

> The only thing that makes me keep wanting to go back to an LSM is the
> fact that the code defining the whitelist seems out of place in the vfs.
> But I guess that's actually separated into a modular cgroup, with the
> actual enforcement built in at the vfs. So that's really the best
> solution.
>
> -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 Pavel Emelianov on Thu, 20 Dec 2007 07:42:24 GMT
View Forum Message <> Reply to Message

Oren Laadan wrote:
>
> Serge E. Hallyn wrote:
>> Quoting Pavel Emelyanov (xemul@openvz.org):
>>> Oren Laadan wrote:
>>>> Serge E. Hallyn wrote:
>>>>> Quoting Oren Laadan (orenl@cs.columbia.edu):
>>>>>> I hate to bring this again, but what if the admin in the container
>>>>>> mounts an external file system (eg. nfs, usb, loop mount from a file,
>>>>>> or via fuse), and that file system already has a device that we would
>>>>>> like to ban inside that container ?
>>>>> Miklos' user mount patches enforced that if !capable(CAP_MKNOD),
>>>>> then mnt->mnt_flags |= MNT_NODEV. So that's no problem.
>>>> Yes, that works to disallow all device files from a mounted file system.
>>>>

>>>> But it's a black and white thing: either they are all banned or allowed;
>>>> you can't have some devices allowed and others not, depending on type
>>>> A scenario where this may be useful is, for instance, if we some apps in
>>>> the container to execute withing a pre-made chroot (sub)tree within that
>>>> container.
>>>>
>>>>> But that's been pulled out of -mm! ? Crap.
>>>>>
>>>>>> Since anyway we will have to keep a white- (or black-) list of devices
>>>>>> that are permitted in a container, and that list may change even change
>>>>>> per container -- why not enforce the access control at the VFS layer ?
>>>>>> It's safer in the long run.
>>>>> By that you mean more along the lines of Pavel's patch than my whitelist
>>>>> LSM, or you actually mean Tetsuo's filesystem (i assume you don't mean that
>>>>> by 'vfs layer' :), or something different entirely?
>>>> :)
>>>>
>>>> By 'vfs' I mean at open() time, and not at mount(), or mknod() time.
>>>> Either yours or Pavel's; I tend to prefer not to use LSM as it may
>>>> collide with future security modules.
>>> Oren, AFAIS you've seen my patches for device access controller, right?
>
> If you mean this one:
> http://openvz.org/pipermail/devel/2007-September/007647.html
> then ack :)

Great! Thanks.

>>> Maybe we can revisit the issue then and try to come to agreement on what
>>> kind of model and implementation we all want?
>> That would be great, Pavel. I do prefer your solution over my LSM, so
>> if we can get an elegant block device control right in the vfs code that
>> would be my preference.
>
> I concur.
>
> So it seems to me that we are all in favor of the model where open()
> of a device will consult a black/white-list. Also, we are all in favor
> of a non-LSM implementation, Pavel's code being a good example.

Thank you, Oren and Serge! I will revisit this issue then, but
I have a vacation the next week and, after this, we have a New
Year and Christmas holidays in Russia. So I will be able to go
on with it only after the 7th January :(Hope this is OK for you.

Besides, Andrew told that he would pay little attention to new
features till the 2.6.24 release, so I'm afraid we won't have this
even in -mm in the nearest months :(

Thanks,
Pavel

> Oren.
>
>> The only thing that makes me keep wanting to go back to an LSM is the
>> fact that the code defining the whitelist seems out of place in the vfs.
>> But I guess that's actually separated into a modular cgroup, with the
>> actual enforcement built in at the vfs. So that's really the best
>> solution.
>>
>> -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 serue on Thu, 20 Dec 2007 14:09:42 GMT
View Forum Message <> Reply to Message

Quoting Pavel Emelyanov (xemul@openvz.org):
> Oren Laadan wrote:
> >
> > Serge E. Hallyn wrote:
> >> Quoting Pavel Emelyanov (xemul@openvz.org):
> >>> Oren Laadan wrote:
> >>>> Serge E. Hallyn wrote:
> >>>>> Quoting Oren Laadan (orenl@cs.columbia.edu):
> >>>>>> I hate to bring this again, but what if the admin in the container
> >>>>>> mounts an external file system (eg. nfs, usb, loop mount from a file,
> >>>>>> or via fuse), and that file system already has a device that we would
> >>>>>> like to ban inside that container ?
> >>>>> Miklos' user mount patches enforced that if !capable(CAP_MKNOD),
> >>>>> then mnt->mnt_flags |= MNT_NODEV. So that's no problem.
> >>>> Yes, that works to disallow all device files from a mounted file system.
> >>>>
> >>>> But it's a black and white thing: either they are all banned or allowed;
> >>>> you can't have some devices allowed and others not, depending on type
> >>>> A scenario where this may be useful is, for instance, if we some apps in
> >>>> the container to execute withing a pre-made chroot (sub)tree within that
> >>>> container.
> >>>>
> >>> But that's been pulled out of -mm! ? Crap.

> >>>>>
> >>>>>> Since anyway we will have to keep a white- (or black-) list of devices
> >>>>>> that are permitted in a container, and that list may change even change
> >>>>>> per container -- why not enforce the access control at the VFS layer ?
> >>>>>> It's safer in the long run.
> >>>>> By that you mean more along the lines of Pavel's patch than my whitelist
> >>>>> LSM, or you actually mean Tetsuo's filesystem (i assume you don't mean that
> >>>>> by 'vfs layer' :), or something different entirely?
> >>>> :)
> >>>>
> >>>> By 'vfs' I mean at open() time, and not at mount(), or mknod() time.
> >>>> Either yours or Pavel's; I tend to prefer not to use LSM as it may
> >>>> collide with future security modules.
> >>> Oren, AFAIS you've seen my patches for device access controller, right?
> >
> > If you mean this one:
> > http://openvz.org/pipermail/devel/2007-September/007647.html
> > then ack :)
>
> Great! Thanks.
>
> >>> Maybe we can revisit the issue then and try to come to agreement on what
> >>> kind of model and implementation we all want?
> >> That would be great, Pavel. I do prefer your solution over my LSM, so
> >> if we can get an elegant block device control right in the vfs code that
> >> would be my preference.
> >
> > I concur.
> >
> > So it seems to me that we are all in favor of the model where open()
> > of a device will consult a black/white-list. Also, we are all in favor
> > of a non-LSM implementation, Pavel's code being a good example.
>
> Thank you, Oren and Serge! I will revisit this issue then, but
> I have a vacation the next week and, after this, we have a New
> Year and Christmas holidays in Russia. So I will be able to go
> on with it only after the 7th January :(Hope this is OK for you.
>
> Besides, Andrew told that he would pay little attention to new
> features till the 2.6.24 release, so I'm afraid we won't have this
> even in -mm in the nearest months :(
>
> Thanks,
> Pavel

Cool, let me know any way I can help when you get started.

thanks,

-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 Oren Laadan on Fri, 21 Dec 2007 01:46:38 GMT
View Forum Message <> Reply to Message

Pavel Emelyanov wrote:
> Oren Laadan wrote:
>> Serge E. Hallyn wrote:
>>> Quoting Pavel Emelyanov (xemul@openvz.org):
>>>> Oren Laadan wrote:
>>>>> Serge E. Hallyn wrote:
>>>>>> Quoting Oren Laadan (orenl@cs.columbia.edu):
>>>>>>> I hate to bring this again, but what if the admin in the container
>>>>>>> mounts an external file system (eg. nfs, usb, loop mount from a file,
>>>>>>> or via fuse), and that file system already has a device that we would
>>>>>>> like to ban inside that container ?
>>>>>> Miklos' user mount patches enforced that if !capable(CAP_MKNOD),
>>>>>> then mnt->mnt_flags |= MNT_NODEV. So that's no problem.
>>>>> Yes, that works to disallow all device files from a mounted file system.
>>>>>
>>>>> But it's a black and white thing: either they are all banned or allowed;
>>>>> you can't have some devices allowed and others not, depending on type
>>>>> A scenario where this may be useful is, for instance, if we some apps in
>>>>> the container to execute withing a pre-made chroot (sub)tree within that
>>>>> container.
>>>>>
>>>>> But that's been pulled out of -mm! ? Crap.
>>>>>
>>>>> Since anyway we will have to keep a white- (or black-) list of devices
>>>>> that are permitted in a container, and that list may change even change
>>>>> per container -- why not enforce the access control at the VFS layer ?
>>>>> It's safer in the long run.
>>>> By that you mean more along the lines of Pavel's patch than my whitelist
>>>> LSM, or you actually mean Tetsuo's filesystem (i assume you don't mean that
>>>> by 'vfs layer' :), or something different entirely?
>>> :)
>>>
>>> By 'vfs' I mean at open() time, and not at mount(), or mknod() time.
>>> Either yours or Pavel's; I tend to prefer not to use LSM as it may
>>> collide with future security modules.
>> Oren, AFAIS you've seen my patches for device access controller, right?
> If you mean this one:

>> http://openvz.org/pipermail/devel/2007-September/007647.html
>> then ack :)
>
> Great! Thanks.
>
>>>> Maybe we can revisit the issue then and try to come to agreement on what
>>>> kind of model and implementation we all want?
>>> That would be great, Pavel. I do prefer your solution over my LSM, so
>>> if we can get an elegant block device control right in the vfs code that
>>> would be my preference.
>> I concur.
>>
>> So it seems to me that we are all in favor of the model where open()
>> of a device will consult a black/white-list. Also, we are all in favor
>> of a non-LSM implementation, Pavel's code being a good example.
>
> Thank you, Oren and Serge! I will revisit this issue then, but
> I have a vacation the next week and, after this, we have a New
> Year and Christmas holidays in Russia. So I will be able to go
> on with it only after the 7th January :(Hope this is OK for you.
>
> Besides, Andrew told that he would pay little attention to new
> features till the 2.6.24 release, so I'm afraid we won't have this
> even in -mm in the nearest months :(

Sounds great ! (as for the delay, it wasn't the highest priority issue
to begin with, so no worries).

Ah.. coincidentally they are celebrated here, too, on the same time :D
Merry Christmas and Happy New Year !

Oren.

>
> Thanks,
> Pavel
>
>> Oren.
>>
>>> The only thing that makes me keep wanting to go back to an LSM is the
>>> fact that the code defining the whitelist seems out of place in the vfs.
>>> But I guess that's actually separated into a modular cgroup, with the
>>> actual enforcement built in at the vfs. So that's really the best
>>> solution.
>>>
>>> -serge
>

_____