Subject: Re: [PATCH] Send quota messages via netlink
Posted by ebiederm on Wed, 29 Aug 2007 21:06:43 GMT
View Forum Message <> Reply to Message

Jan Kara <jack@suse.cz> writes:

>
>> However I'm still confused about the use of current->user. If that
>> is what we really want and not the user who's quota will be charged
>> it gets to be a really trick business, because potentially the uid
>> we want to deliver varies depending on who opened the netlink socket.
>   I see it's a complicated matter :). What I need to somehow pass to
> userspace is something (and I don't really care whether it will be number,
> string or whatever) that userspace can read and e.g. find a terminal
> window or desktop the affected user has open and also translate the
> identity to some user-understandable name (average user Joe has to
> understand that he should quickly cleanup his home directory ;).
>   Thinking more about it, we could probably pass a string to userspace in
> the format:
>   <namespace type>:<user identification>
>
> So for example we can have something like:
>   unix:1000 (traditional unix UIDs)
>   nfs4:joe@machine
>
> The problem is: Are we able to find out in which "namespace type" we are
> and send enough identifying information from a context of unpriviledged
> user?

Ok. This provides enough context to understand what you are trying to do.
You do want the unix user id, not the filesystem notion. Because you
are looking for the user.

So we have to figure out how to do the hard thing which is look at
who opened our netlink broadcast see if they are in the same user
namespace as current->user. Which is a pain and we don't currently
have the infrastructure for.

Eric

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

Subject: Re: [PATCH] Send quota messages via netlink
Posted by Valdis.Kletnieks on Wed, 29 Aug 2007 21:19:42 GMT

View Forum Message <> Reply to Message

On Wed, 29 Aug 2007 15:06:43 MDT, Eric W. Biederman said:

> So we have to figure out how to do the hard thing which is look at
> who opened our netlink broadcast see if they are in the same user
> namespace as current->user. Which is a pain and we don't currently
> have the infrastructure for.

Provision also needs to be made for things that are listening to the
netlink broadcasts that don't match the user doing the operation or
the owner of the file - similar to the way auditd wants events.

Subject: Re: [PATCH] Send quota messages via netlink
Posted by Jan Kara on Thu, 30 Aug 2007 09:25:48 GMT
View Forum Message <> Reply to Message

On Wed 29-08-07 15:06:43, Eric W. Biederman wrote:
> Jan Kara <jack@suse.cz> writes:
> >> However I'm still confused about the use of current->user. If that
> >> is what we really want and not the user who's quota will be charged
> >> it gets to be a really trick business, because potentially the uid
> >> we want to deliver varies depending on who opened the netlink socket.
> >   I see it's a complicated matter :). What I need to somehow pass to
> > userspace is something (and I don't really care whether it will be number,
> > string or whatever) that userspace can read and e.g. find a terminal
> > window or desktop the affected user has open and also translate the
> > identity to some user-understandable name (average user Joe has to
> > understand that he should quickly cleanup his home directory ;).
> >   Thinking more about it, we could probably pass a string to userspace in
> > the format:
> >   <namespace type>:<user identification>
> >
> > So for example we can have something like:
> >   unix:1000 (traditional unix UIDs)
> >   nfs4:joe@machine
> >
> > The problem is: Are we able to find out in which "namespace type" we are
> > and send enough identifying information from a context of unpriviledged
> > user?
>

> Ok. This provides enough context to understand what you are trying to do.
> You do want the unix user id, not the filesystem notion. Because you
> are looking for the user.
>
> So we have to figure out how to do the hard thing which is look at
> who opened our netlink broadcast see if they are in the same user
> namespace as current->user. Which is a pain and we don't currently
> have the infrastructure for.
  There can be arbitrary number of listeners (potentially from different
namespaces if I understand it correctly) listening to broadcasts. So I
think we should pass some universal identifier rather than try to find out
who is listening etc. I think such identifiers would be useful for other
things too, won't they?
  BTW: Do you have some idea, when would be the infrastructure clearer?
Whether it makes sence to currently proceed with UIDs and later change it
to something generic or whether I should wait before you sort it out :).

        Honza
--
Jan Kara <jack@suse.cz>
SuSE CR Labs

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

---

Subject: Re: [PATCH] Send quota messages via netlink
Posted by ebiederm on Thu, 30 Aug 2007 17:33:09 GMT
View Forum Message <> Reply to Message

Jan Kara <jack@suse.cz> writes:
>   There can be arbitrary number of listeners (potentially from different
> namespaces if I understand it correctly) listening to broadcasts. So I
> think we should pass some universal identifier rather than try to find out
> who is listening etc. I think such identifiers would be useful for other
> things too, won't they?

So internal to the kernel we have such a universal identifier.
struct user.

There are to practical questions.
1) How do we present that information to user space?
2) How does user space want to process this information?

If we only want user space to be able to look up a user and send
him a message. It probably makes sense to do the struct user to
uid conversion in the proper context in the kernel because we have

that information.

If this is a general feature that happens to allows us to look up
the user given the filesystems view of what is going on would be
easier in the kernel, and not require translation. But it means
that we can't support 9p and nfs for now. But since we don't support
quotas on the client end anyway that doesn't sound like a big deal.

The problem with the filesystem view is that there will be occasions
where we simply can not map a user into it, because the filesystem
won't have a concept of that particular user.

So we could run into the situation where alice owns the file. Bob
writes to the file and pushes it over quota. But the filesystem
has no concept of who bob is. So we won't be able to report that
it was bob that pushed things over the edge.

> BTW: Do you have some idea, when would be the infrastructure clearer?

So the plan is to get to the point where are uid comparisons in the
kernel are (user namespace, uid) comparisons. Or possibly struct
user comparisons (depending on the context. And struct mount will
contain the user namespace of whoever mounted the filesystem.

Adding infrastructure to netlink to allow us to do conversions
as the packets are enqueued for a specific user is something I
would rather avoid, but that is a path we can go down if we have
to.

> Whether it makes sence to currently proceed with UIDs and later change it
> to something generic or whether I should wait before you sort it out :).

A good question. I think things are clear enough that it at least
makes sense to sketch a solution to the problem even if we don't
implement it at this point.

I have been hoping Cedric or Serge would jump in because I think those
are the guys who have been working on the implementation.

Eric

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

---

Subject: Re: [PATCH] Send quota messages via netlink

Quoting Eric W. Biederman (ebiederm@xmission.com):
> Jan Kara <jack@suse.cz> writes:
> > There can be arbitrary number of listeners (potentially from different
> > namespaces if I understand it correctly) listening to broadcasts. So I
> > think we should pass some universal identifier rather than try to find out
> > who is listening etc. I think such identifiers would be useful for other
> > things too, won't they?
>
> So internal to the kernel we have such a universal identifier.
> struct user.
>
> There are to practical questions.
> 1) How do we present that information to user space?
> 2) How does user space want to process this information?
>
> If we only want user space to be able to look up a user and send
> him a message. It probably makes sense to do the struct user to
> uid conversion in the proper context in the kernel because we have
> that information.
>
> If this is a general feature that happens to allows us to look up
> the user given the filesystems view of what is going on would be
> easier in the kernel, and not require translation. But it means
> that we can't support 9p and nfs for now. But since we don't support
> quotas on the client end anyway that doesn't sound like a big deal.
>
> The problem with the filesystem view is that there will be occasions
> where we simply can not map a user into it, because the filesystem
> won't have a concept of that particular user.
>
> So we could run into the situation where alice owns the file. Bob
> writes to the file and pushes it over quota. But the filesystem
> has no concept of who bob is. So we won't be able to report that
> it was bob that pushed things over the edge.
>
> > BTW: Do you have some idea, when would be the infrastructure clearer?
>
> So the plan is to get to the point where are uid comparisons in the
> kernel are (user namespace, uid) comparisons. Or possibly struct
> user comparisons (depending on the context. And struct mount will
> contain the user namespace of whoever mounted the filesystem.
>
> Adding infrastructure to netlink to allow us to do conversions
> as the packets are enqueued for a specific user is something I
> would rather avoid, but that is a path we can go down if we have
> to.

>
> > Whether it makes sence to currently proceed with UIDs and later change it
> > to something generic or whether I should wait before you sort it out :).
>
> A good question. I think things are clear enough that it at least
> makes sense to sketch a solution to the problem even if we don't
> implement it at this point.
>
> I have been hoping Cedric or Serge would jump in because I think those
> are the guys who have been working on the implementation.

Sorry, I've lost the original patch from two separate mailboxes...

The proper behavior depends on how we end up tying filesystems to user namespaces, which isn't actually decided yet.

The way I was recommending doing that was:

A filesystem is tied to a user namespace. If a uid in another naemspace is to be allowed to access the filesystem, it will actually - through a key in it's keyring (which acts like a capability) - be mapped to a uid in the filesystem's uid namespace. So in Eric's example, if Alice brings Bob over quota, Alice would have done so through some user Charlie who she is authorized to act as through her keyring. So Charlie should be the id which would be logged over netlink.

Of course there is currently no support for this. So I'd recommend one of two options: either just punt on uid namespace for now and we'll fix it when we improve user namespaces - so log Alice's userid. Or we can try to do it somewhat correct now, which might be done as follows:

1. introduce get_uid_in_userns(tsk). For now this just returns
   tsk->uid if current->userns == tsk->userns, else it returns
   0.
   This way in Eric's scenario, Bob would be told that root,
   not an invalid user (Alice) had brought him over quota.
   Eventually, this would walk tsk's keychain for a uid entry
   in current's active user namespace.

2. Add the userns to the netlink message.

Again I need to find Jan's orginal patch, but I'll take a look at this.

-serge

Subject: Re: [PATCH] Send quota messages via netlink
Posted by serge on Thu, 30 Aug 2007 19:10:10 GMT
View Forum Message <> Reply to Message

Quoting Jan Kara (jack@suse.cz):
> On Wed 29-08-07 15:06:43, Eric W. Biederman wrote:
> > Jan Kara <jack@suse.cz> writes:
> > >> However I'm still confused about the use of current->user. If that
> > >> is what we really want and not the user who's quota will be charged
> > >> it gets to be a really trick business, because potentially the uid
> > >> we want to deliver varies depending on who opened the netlink socket.
> > >   I see it's a complicated matter :). What I need to somehow pass to
> > > userspace is something (and I don't really care whether it will be number,
> > > string or whatever) that userspace can read and e.g. find a terminal
> > > window or desktop the affected user has open and also translate the
> > > identity to some user-understandable name (average user Joe has to
> > > understand that he should quickly cleanup his home directory ;).
> > >   Thinking more about it, we could probably pass a string to userspace in
> > > the format:
> > >   <namespace type>:<user identification>
> > >
> > > So for example we can have something like:
> > >   unix:1000 (traditional unix UIDs)
> > >   nfs4:joe@machine
> > >
> > > The problem is: Are we able to find out in which "namespace type" we are
> > > and send enough identifying information from a context of unpriviledged
> > > user?
> >
> > Ok. This provides enough context to understand what you are trying to do.
> > You do want the unix user id, not the filesystem notion. Because you
> > are looking for the user.
> >
> > So we have to figure out how to do the hard thing which is look at
> > who opened our netlink broadcast see if they are in the same user
> > namespace as current->user. Which is a pain and we don't currently
> > have the infrastructure for.
>   There can be arbitrary number of listeners (potentially from different
> namespaces if I understand it correctly) listening to broadcasts. So I

Currently that is true, but i think isolating netlink sockets is going
to have to be done pretty soon.

On the one hand cloning a new netlink socket ns when you unshare
CLONE_NEWNET may seem 'obvious', but I think doing so when you unshare
CLONE_NEWUSER make much more sense considering netlink's use for audit
and now for quota.

> think we should pass some universal identifier rather than try to find out

Even with isolating netlink we still may want to send out an identifier.
However, just as with mounts extensions we're printing out the memory
address of vfsmounts, we might just want to print out the memory address
of the userns. It's not universal, but should be good enough.

-serge

> who is listening etc. I think such identifiers would be useful for other
> things too, won't they?
>   BTW: Do you have some idea, when would be the infrastructure clearer?
> Whether it makes sence to currently proceed with UIDs and later change it
> to something generic or whether I should wait before you sort it out :).
>
>   Honza
> --
> Jan Kara <jack@suse.cz>
> SuSE CR Labs
> -
> 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/

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

---

Subject: Re: [PATCH] Send quota messages via netlink
Posted by serge on Thu, 30 Aug 2007 19:18:09 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> Jan Kara <jack@suse.cz> writes:
> >   There can be arbitrary number of listeners (potentially from different
> > namespaces if I understand it correctly) listening to broadcasts. So I
> > think we should pass some universal identifier rather than try to find out
> > who is listening etc. I think such identifiers would be useful for other
> > things too, won't they?
>
> So internal to the kernel we have such a universal identifier.
> struct user.
>
> There are to practical questions.
> 1) How do we present that information to user space?
> 2) How does user space want to process this information?
>

> If we only want user space to be able to look up a user and send
> him a message. It probably makes sense to do the struct user to
> uid conversion in the proper context in the kernel because we have
> that information.
>
> If this is a general feature that happens to allows us to look up
> the user given the filesystems view of what is going on would be
> easier in the kernel, and not require translation. But it means
> that we can't support 9p and nfs for now. But since we don't support
> quotas on the client end anyway that doesn't sound like a big deal.
>
> The problem with the filesystem view is that there will be occasions
> where we simply can not map a user into it, because the filesystem
> won't have a concept of that particular user.
>
> So we could run into the situation where alice owns the file. Bob
> writes to the file and pushes it over quota. But the filesystem
> has no concept of who bob is. So we won't be able to report that
> it was bob that pushed things over the edge.
>
> > BTW: Do you have some idea, when would be the infrastructure clearer?
>
> So the plan is to get to the point where are uid comparisons in the
> kernel are (user namespace, uid) comparisons. Or possibly struct

Just fyi Eric,

Note that given the amount of churn going on due to pid and network
namespaces, I was seeing completion of user namespaces as something to
be done sometime next year. In the meantime I was only going to do
something with capabilities to restrict root in user namespaces (which I
think will take the form of per-process non-expandable cap_bsets, which
I plan to start basically right now).

But I'll gladly do the userns enhancements earlier if it's actually
wanted. They promise to be great fun :)

-serge

> user comparisons (depending on the context. And struct mount will
> contain the user namespace of whoever mounted the filesystem.
>
> Adding infrastructure to netlink to allow us to do conversions
> as the packets are enqueued for a specific user is something I
> would rather avoid, but that is a path we can go down if we have
> to.
>
> > Whether it makes sence to currently proceed with UIDs and later change it

> > to something generic or whether I should wait before you sort it out :).
>
> A good question. I think things are clear enough that it at least
> makes sense to sketch a solution to the problem even if we don't
> implement it at this point.
>
> I have been hoping Cedric or Serge would jump in because I think those
> are the guys who have been working on the implementation.
>
> Eric
> -
> 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/

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

---

Subject: Re: [PATCH] Send quota messages via netlink
Posted by Jan Kara on Thu, 30 Aug 2007 22:18:25 GMT
View Forum Message <> Reply to Message

On Thu 30-08-07 14:10:10, Serge E. Hallyn wrote:
> Quoting Jan Kara (jack@suse.cz):
> > On Wed 29-08-07 15:06:43, Eric W. Biederman wrote:
> > > Jan Kara <jack@suse.cz> writes:
> > > >> However I'm still confused about the use of current->user. If that
> > > >> is what we really want and not the user who's quota will be charged
> > > >> it gets to be a really trick business, because potentially the uid
> > > >> we want to deliver varies depending on who opened the netlink socket.
> > > >   I see it's a complicated matter :). What I need to somehow pass to
> > > > userspace is something (and I don't really care whether it will be number,
> > > > string or whatever) that userspace can read and e.g. find a terminal
> > > > window or desktop the affected user has open and also translate the
> > > > identity to some user-understandable name (average user Joe has to
> > > > understand that he should quickly cleanup his home directory ;).
> > > >   Thinking more about it, we could probably pass a string to userspace in
> > > > the format:
> > > >   <namespace type>:<user identification>
> > > >
> > > > So for example we can have something like:
> > > >   unix:1000 (traditional unix UIDs)
> > > >   nfs4:joe@machine
> > > >
> > > > The problem is: Are we able to find out in which "namespace type" we are

> > > > and send enough identifying information from a context of unpriviledged
> > > > user?
> > > >
> > > Ok. This provides enough context to understand what you are trying to do.
> > > You do want the unix user id, not the filesystem notion. Because you
> > > are looking for the user.
> > > >
> > > So we have to figure out how to do the hard thing which is look at
> > > who opened our netlink broadcast see if they are in the same user
> > > namespace as current->user. Which is a pain and we don't currently
> > > have the infrastructure for.
> >   There can be arbitrary number of listeners (potentially from different
> > namespaces if I understand it correctly) listening to broadcasts. So I
>
> Currently that is true, but i think isolating netlink sockets is going
> to have to be done pretty soon.
>
> On the one hand cloning a new netlink socket ns when you unshare
> CLONE_NEWNET may seem 'obvious', but I think doing so when you unshare
> CLONE_NEWUSER make much more sense considering netlink's use for audit
> and now for quota.
>
> > think we should pass some universal identifier rather than try to find out
>
> Even with isolating netlink we still may want to send out an identifier.
> However, just as with mounts extensions we're printing out the memory
> address of vfsmounts, we might just want to print out the memory address
> of the userns. It's not universal, but should be good enough.

Maybe before proceeding further with the discussion I'd like to understand following: What are these user namespaces supposed to be good for?

I imagine it so that you have a machine and on it several virtual machines which are sharing a filesystem (or it could be a cluster). Now you want UIDs to be independent between these virtual machines. That's it, right?

Now to continue the example: Alice has UID 100 on machineA, Bob has UID 100 on machineB. These translate to UIDs 1000 and 1001 on the common filesystem. Process of Alice writes to a file and Bob becomes to be over quota. In this situation, there would be probably two processes (from machineA and machineB) listening on the netlink socket. We want to send a message so that on Alice's desktop we can show a message: "You caused Bob to exceed his quotas" and of Bob's desktop: "Alice has caused that you are over quota.".

Because there may be is not a notion of Bob on machineA or of Alice on machineB, we are in trouble, right? What I like the most is to use the filesystem identities (as you suggested in some other email). I. e. because both Alice and Bob share a filesystem, identities of both have to make sense to it (for example for purposes of permission checking). So we can probably

send via netlink these (in our example ids 1000 and 1001) and hope that inside machineA and machineB there will be a way to translate these identities to names "Alice" and "Bob". So that user can understand what is happenning. Does this sound plausible?
  If we go this route, then we only need a kernel function, that will for a pair ($filesystem, $task) return indentity of that $task used for operations on $filesystem...

Honza
--
Jan Kara <jack@suse.cz>
SuSE CR Labs

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