Subject: cryo and mm->arg_start
Posted by serue on Fri, 11 Jul 2008 13:13:45 GMT
View Forum Message <> Reply to Message

What cryo does right now to restart some task (say openmp stream) is:

1. fork, ptrace_tracem(), then execute the original application (stream)
2. (some other stuff)
3. through ptrace, cause the restarted process to read the checkpointed data back into writeable maps. This includes the stack

The restarted task's filename is correctly reported through /proc/$$/cmdline. Once we rewrite the stack, it is corrupted.

The reason is that the cmdline contents are taken from mm->arg_start, which varies with each execution.

On the one hand it's kind of a "small thing." But IIUC it's like did_exec in that there is no way to fix it for userspace.

One thing we could do here is to start extending the cryo approach with Eric's checkpoint-as-a-coredump (caac?). We generate the tiniest of coredumps which, at first, contains nothing but mm->arg_start and maybe a process id. It would be simplest if it also contained a filename for the real executable, but I don't know that we could get away with that. If we *could* get away with that, then we could have a trivial fs/binfmt_cr.c "execute" such a caac file, which would mean it would exec the original executable, then change process settings in accordance with the ccac file contents.

Any other ideas? Comments?

-serge

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

Subject: Re: cryo and mm->arg_start
Posted by Dave Hansen on Fri, 11 Jul 2008 16:38:30 GMT
View Forum Message <> Reply to Message

On Fri, 2008-07-11 at 08:13 -0500, Serge E. Hallyn wrote:
>

> One thing we could do here is to start extending the cryo approach
> with Eric's checkpoint-as-a-coredump (caac?).  We generate the
> tiniest of coredumps which, at first, contains nothing but
> mm->arg_start and maybe a process id.  It would be simplest if
> it also contained a filename for the real executable,

The exec model sounds reasonable to me.

But, I think the filename of the exe is going to have to be in the checkpoint *already*.  It is mapped by at least one of the VMAs, and will probably be dumped as a normal file-backed area.

Now, since arg_start is already set up at exec time, it doesn't seem unreasonable to have the theoretical fs/binfmt_cr.c set it as well.

-- Dave

_______________________________________________

Subject: Re: cryo and mm->arg_start
Posted by serue on Fri, 11 Jul 2008 21:26:39 GMT
View Forum Message <> Reply to Message

Quoting Dave Hansen (dave@linux.vnet.ibm.com):
> On Fri, 2008-07-11 at 08:13 -0500, Serge E. Hallyn wrote:
> >
> > One thing we could do here is to start extending the cryo approach
> > with Eric's checkpoint-as-a-coredump (caac?).  We generate the
> > tiniest of coredumps which, at first, contains nothing but
> > mm->arg_start and maybe a process id.  It would be simplest if
> > it also contained a filename for the real executable,
>
> The exec model sounds reasonable to me.
>
> But, I think the filename of the exe is going to have to be in the
> checkpoint *already*.  It is mapped by at least one of the VMAs, and
> will probably be dumped as a normal file-backed area.
>
> Now, since arg_start is already set up at exec time, it doesn't seem
> unreasonable to have the theoretical fs/binfmt_cr.c set it as well.
>
> -- Dave

Ok.

So I'll play with this a bit over the next week.  I'm mostly unfamiliar
with the coredump code and have looked through the binfmts mainly for
tracking the order of security events, so this should be fun.

-serge

## Subject: Re: cryo and mm->arg_start
## Posted by Matt Helsley on Fri, 11 Jul 2008 22:01:13 GMT

View Forum Message <> Reply to Message

On Fri, 2008-07-11 at 09:38 -0700, Dave Hansen wrote:
> On Fri, 2008-07-11 at 08:13 -0500, Serge E. Hallyn wrote:
> >
> > One thing we could do here is to start extending the cryo approach
> > with Eric's checkpoint-as-a-coredump (caac?).  We generate the
> > tiniest of coredumps which, at first, contains nothing but
> > mm->arg_start and maybe a process id.  It would be simplest if
> > it also contained a filename for the real executable,
>
> The exec model sounds reasonable to me.
>
> But, I think the filename of the exe is going to have to be in the
> checkpoint *already*.  It is mapped by at least one of the VMAs, and
> will probably be dumped as a normal file-backed area.

 Yes, the file that backed the exec will be there. Note that thanks to
"stacking" filesystems the path to the file backing the exe is not
_always_ going to be the same as the path to the file which userspace
exec'd in the first place. You can see this by comparing
the /proc/<pid>/exe symlink with the file backing the VMA.

 This is important to any program which checks the /proc/self/exe
symlink to find out where it's installed (Java does this, for example).
I think it's possible to do this with a binfmt -- it's just one more
detail to remember.

Cheers,
 -Matt

https://lists.linux-foundation.org/mailman/listinfo/containers

Subject: Re: cryo and mm->arg_start
Posted by serue on Sun, 13 Jul 2008 21:08:46 GMT
View Forum Message <> Reply to Message

Quoting Matt Helsley (matthltc@us.ibm.com):

```
>
> On Fri, 2008-07-11 at 09:38 -0700, Dave Hansen wrote:
> > On Fri, 2008-07-11 at 08:13 -0500, Serge E. Hallyn wrote:
> > >
> > > One thing we could do here is to start extending the cryo approach
> > > with Eric's checkpoint-as-a-coredump (caac?).  We generate the
> > > tiniest of coredumps which, at first, contains nothing but
> > > mm->arg_start and maybe a process id.  It would be simplest if
> > > it also contained a filename for the real executable,
> >
> > The exec model sounds reasonable to me.
> >
> > But, I think the filename of the exe is going to have to be in the
> > checkpoint *already*.  It is mapped by at least one of the VMAs, and
> > will probably be dumped as a normal file-backed area.
>
>  Yes, the file that backed the exec will be there. Note that thanks to
> "stacking" filesystems the path to the file backing the exe is not
> _always_ going to be the same as the path to the file which userspace
> exec'd in the first place. You can see this by comparing
> the /proc/<pid>/exe symlink with the file backing the VMA.
>
>  This is important to any program which checks the /proc/self/exe
> symlink to find out where it's installed (Java does this, for example).
> I think it's possible to do this with a binfmt -- it's just one more
> detail to remember.
>
> Cheers,
>  -Matt
```

Let's say that before starting my checkpointable job, I did

```
 mount -t ecryptfs /home/hallyn /home/hallyn
```

Now if the checkpointable job is /home/hallyn/somelongjob, then I think
it's fair to say that restart can fail if /home/hallyn at the restart
machine isn't ecryptfs-mounted.

In that case, would you still think there is a problem?

On the other hand, if the checkpointable job did the ecryptfs mount
itself, then it would be expected that at restart the ecryptfs mount
would be remounted.  How that would be done I have no idea offhand.

thanks,
-serge

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

---

Subject: Re: cryo and mm->arg_start
Posted by Sukadev Bhattiprolu on Tue, 15 Jul 2008 21:40:50 GMT
View Forum Message <> Reply to Message

Serge E. Hallyn [serue@us.ibm.com] wrote:
| Quoting Matt Helsley (matthltc@us.ibm.com):
| >
| > On Fri, 2008-07-11 at 09:38 -0700, Dave Hansen wrote:
| > > On Fri, 2008-07-11 at 08:13 -0500, Serge E. Hallyn wrote:
| > > >
| > > > One thing we could do here is to start extending the cryo approach
| > > > with Eric's checkpoint-as-a-coredump (caac?).  We generate the
| > > > tiniest of coredumps which, at first, contains nothing but
| > > > mm->arg_start and maybe a process id.  It would be simplest if
| > > > it also contained a filename for the real executable,
| > >
| > > The exec model sounds reasonable to me.
| > >
| > > But, I think the filename of the exe is going to have to be in the
| > > checkpoint *already*.  It is mapped by at least one of the VMAs, and
| > > will probably be dumped as a normal file-backed area.
| >
| >  Yes, the file that backed the exec will be there. Note that thanks to
| > "stacking" filesystems the path to the file backing the exe is not
| > _always_ going to be the same as the path to the file which userspace
| > exec'd in the first place. You can see this by comparing
| > the /proc/<pid>/exe symlink with the file backing the VMA.
| >
| >  This is important to any program which checks the /proc/self/exe
| > symlink to find out where it's installed (Java does this, for example).
| > I think it's possible to do this with a binfmt -- it's just one more
| > detail to remember.
| >
| > Cheers,
| >  -Matt
|

| Let's say that before starting my checkpointable job, I did
|
|  mount -t ecryptfs /home/hallyn /home/hallyn
|
| Now if the checkpointable job is /home/hallyn/somelongjob, then I think
| it's fair to say that restart can fail if /home/hallyn at the restart
| machine isn't ecryptfs-mounted.
|
| In that case, would you still think there is a problem?
|
| On the other hand, if the checkpointable job did the ecryptfs mount
| itself, then it would be expected that at restart the ecryptfs mount
| would be remounted.  How that would be done I have no idea offhand.

Hmm, wonder if the new /proc/pid/mountinfo with its mount-ids would
enable us to identify the filesystems that a given process expects.

Which brings up another question. If two processes in the same container
have different mount namespaces and mount points, we would need to
reestablish the mounts during restart right ?

Suka

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

## Subject: Re: cryo and mm->arg_start
## Posted by serue on Wed, 16 Jul 2008 15:23:05 GMT

View Forum Message <> Reply to Message

Quoting sukadev@us.ibm.com (sukadev@us.ibm.com):
> Serge E. Hallyn [serue@us.ibm.com] wrote:
> | Quoting Matt Helsley (matthltc@us.ibm.com):
> | >
> | > On Fri, 2008-07-11 at 09:38 -0700, Dave Hansen wrote:
> | > > On Fri, 2008-07-11 at 08:13 -0500, Serge E. Hallyn wrote:
> | > > >
> | > > > One thing we could do here is to start extending the cryo approach
> | > > > with Eric's checkpoint-as-a-coredump (caac?).  We generate the
> | > > > tiniest of coredumps which, at first, contains nothing but
> | > > > mm->arg_start and maybe a process id.  It would be simplest if
> | > > > it also contained a filename for the real executable,
> | > >
> | > > The exec model sounds reasonable to me.
> | > >
> | > > But, I think the filename of the exe is going to have to be in the

> | > > checkpoint *already*.  It is mapped by at least one of the VMAs, and
> | > > will probably be dumped as a normal file-backed area.
> | >
> | >  Yes, the file that backed the exec will be there. Note that thanks to
> | > "stacking" filesystems the path to the file backing the exe is not
> | > _always_ going to be the same as the path to the file which userspace
> | > exec'd in the first place. You can see this by comparing
> | > the /proc/<pid>/exe symlink with the file backing the VMA.
> | >
> | >  This is important to any program which checks the /proc/self/exe
> | > symlink to find out where it's installed (Java does this, for example).
> | > I think it's possible to do this with a binfmt -- it's just one more
> | > detail to remember.
> | >
> | > Cheers,
> | >  -Matt
> |
> | Let's say that before starting my checkpointable job, I did
> |
> |  mount -t ecryptfs /home/hallyn /home/hallyn
> |
> | Now if the checkpointable job is /home/hallyn/somelongjob, then I think
> | it's fair to say that restart can fail if /home/hallyn at the restart
> | machine isn't ecryptfs-mounted.
> |
> | In that case, would you still think there is a problem?
> |
> | On the other hand, if the checkpointable job did the ecryptfs mount
> | itself, then it would be expected that at restart the ecryptfs mount
> | would be remounted.  How that would be done I have no idea offhand.
>
> Hmm, wonder if the new /proc/pid/mountinfo with its mount-ids would
> enable us to identify the filesystems that a given process expects.

Interesting point.  Yes, it *should*, that's sort of the idea.  I don't
remember whether some of the limitations in terms of hiding mount-ids
from other namespaces were implemented or not, if so I suspect they
could be a problem.

> Which brings up another question. If two processes in the same container
> have different mount namespaces and mount points, we would need to
> reestablish the mounts during restart right ?

Yes.

-serge
_______________________________________________
Containers mailing list

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