Arvados - Bug #2246

Race condition in crunch-job: multiple jobs use the same /tmp/crunch-job/ on head node to check out git trees and make archives. Fix as simple as using git archive --remote?

02/28/2014 04:45 PM - Tom Clegg

Status:	Resolved	**Start date:**	04/08/2014
Priority:	Normal	**Due date:**	
Assigned To:	Tom Clegg	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2014-04-16 Dev tools and data/resource management		

Description

Subtasks:

Task # 2521: Use git in --remote mode instead of checking out repository on head node	**Resolved**
Task # 2564: Review 2246-fix-git-clone-race	**Resolved**

Associated revisions

Revision 45ff1f04 - 04/11/2014 12:59 PM - Tom Clegg

Merge branch '2246-fix-git-clone-race'

closes #2246

History

#1 - 03/12/2014 01:06 PM - Tom Clegg

- Release deleted (6)

#2 - 03/12/2014 01:07 PM - Tom Clegg

- Target version set to 2014-04-16 Dev tools and data/resource management

#3 - 03/13/2014 12:40 PM - Tom Clegg

- Story points set to 1.0

#4 - 03/26/2014 04:31 PM - Tom Clegg

- Assigned To set to Tom Clegg

#5 - 03/31/2014 11:32 PM - Tom Clegg

- Status changed from New to Resolved

#6 - 03/31/2014 11:32 PM - Tom Clegg

- Status changed from Resolved to New

#7 - 04/09/2014 05:37 PM - Brett Smith

This is probably out of scope for the branch, but I'll go on the record here: crunch-job has a number of potential shell exploits. Any backticks that interpolate user data are a likely risk. That may not seem like a big deal when the point of crunch-job is to run the user's arbitrary code anyway, but that means it's a good channel for privilege escalation attacks. An attacker who can *only* submit Jobs to the API server can leverage that power to get crunch-job to run arbitrary shell. I think this is serious enough to be worth fixing, especially since the fix is relatively straightforward (use multi-argument open() calls to build pipes, or \Q$var\E in backticks). This branch does make that job easier by eliminating the need to use a shell for these git calls (no cd &&), so thanks for that.

my $repo = $git_dir || $ENV{'CRUNCH_DEFAULT_GIT_DIR'} || $Job->{'repository'};

Are you sure adding the repository field here will do what you mean? The rest of crunch-job is expecting $repo to be a full path, but I think the repository field just has the name. Did I miss some magic somewhere?

All the other changes look good to me. Thanks.

#8 - 04/10/2014 11:50 PM - Tom Clegg

Brett Smith wrote:

> This is probably out of scope for the branch, but I'll go on the record here: crunch-job has a number of potential shell exploits. Any backticks that interpolate user data are a likely risk. That may not seem like a big deal when the point of crunch-job is to run the user's arbitrary code anyway, but that means it's a good channel for privilege escalation attacks. An attacker who can *only* submit Jobs to the API server can leverage that power to get crunch-job to run arbitrary shell. I think this is serious enough to be worth fixing, especially since the fix is relatively straightforward (use multi-argument open() calls to build pipes, or \Q$var\E in backticks). This branch does make that job easier by eliminating the need to use a shell for these git calls (no cd &&), so thanks for that.

Yes, there are lots of assumptions in there. Putting a bunch of \Q\E in there sounds like a good idea. I poked around and added a few obvious ones.

Unfortunately \Q\E (like PHP's escapeshellarg) doesn't seem to consider the "empty string" case: `foo --bar \Q$baz\E --qux` probably doesn't do what you want if $baz eq ''. So I suppose the idiom is `foo --bar "\Q$baz\E --qux` ...? (ouch)

> my $repo = $git_dir || $ENV{'CRUNCH_DEFAULT_GIT_DIR'} || $Job->{'repository'};
>
> Are you sure adding the repository field here will do what you mean? The rest of crunch-job is expecting $repo to be a full path, but I think the repository field just has the name. Did I miss some magic somewhere?

Indeed, this is a bit mysterious. The last resort lets you specify a local path when running a "local" job in your shell VM. Added comments to explain.

#9 - 04/11/2014 10:35 AM - Brett Smith

> Unfortunately \Q\E (like PHP's escapeshellarg) doesn't seem to consider the "empty string" case: `foo --bar \Q$baz\E --qux` probably doesn't do what you want if $baz eq ''. So I suppose the idiom is `foo --bar "\Q$baz\E --qux` ...? (ouch)

I guess so. This is another reason to prefer multi-argument open() calls over anything that interpolates shell. But I think your current implementation works.

Thanks for expounding on the $Job->{repository} thing. I'd forgotten about the local directory use case (which is funny since I was using it to work on Docker crunch jobs). I think this is good to merge. Thanks again.

#10 - 04/11/2014 12:59 PM - Tom Clegg

- Status changed from New to Resolved