

Calamari - Fix #7691

api/v2/<id>/log throws EauthAuthenticationError

03/11/2014 10:33 PM - Dan Mick

Status:	Resolved	% Done:	0%
Priority:	Normal	Spent time:	0.00 hour
Assignee:	Dan Mick		
Category:	Backend (REST API)		
Target version:	v1.2-dev7		
Source:	Development	Affected Versions:	
Tags:		ceph-qa-suite:	
Backport:		Crash signature (v1):	
Reviewed:		Crash signature (v2):	
Description			
From the manage UI on mira106, Tools/Log Tail tries api/v2/<id>/log, and gets an exception:			
<pre>2014-03-12 00:31:11,725 - ERROR - django.request Internal Server Error: /api/v2/cluster/61723e0f-992b-466e-9c09-9914931bc584/log Traceback (most recent call last): File "/opt/calamari/venv/lib/python2.7/site-packages/django/core/handlers/base.py", line 115, in get_response response = callback(request, *callback_args, **callback_kwargs) File "/opt/calamari/venv/lib/python2.7/site-packages/rest_framework/viewsets.py", line 78, in view return self.dispatch(request, *args, **kwargs) File "/opt/calamari/venv/lib/python2.7/site-packages/calamari_rest_api-0.1-py2.7.egg/calamari_rest/views/rpc_view.py", line 38, in dispatch return super(RPCView, self).dispatch(request, *args, **kwargs) File "/opt/calamari/venv/lib/python2.7/site-packages/django/views/decorators/csrf.py", line 77, in wrapped_view return view_func(*args, **kwargs) File "/opt/calamari/venv/lib/python2.7/site-packages/rest_framework/views.py", line 399, in dispatch response = self.handle_exception(exc) File "/opt/calamari/venv/lib/python2.7/site-packages/calamari_rest_api-0.1-py2.7.egg/calamari_rest/views/rpc_view.py", line 52, in handle_exception return super(RPCView, self).handle_exception(exc) File "/opt/calamari/venv/lib/python2.7/site-packages/rest_framework/views.py", line 396, in dispatch response = handler(request, *args, **kwargs) File "/opt/calamari/venv/lib/python2.7/site-packages/calamari_rest_api-0.1-py2.7.egg/calamari_rest/views/v2.py", line 616, in get_cluster_log results = client.cmd(mon_fqdn, "log_tail.tail", ["ceph/{name}.log".format(name=name), lines]) File "/usr/lib/pymodules/python2.7/salt/client/__init__.py", line 517, in cmd **kwargs) File "/usr/lib/pymodules/python2.7/salt/client/__init__.py", line 276, in run_job return self._check_pub_data(pub_data) File "/usr/lib/pymodules/python2.7/salt/client/__init__.py", line 213, in _check_pub_data 'Failed to authenticate, is this user permitted to execute ' EauthAuthenticationError: Failed to authenticate, is this user permitted to execute commands?</pre>			
Related issues:			
Related to Calamari - Fix #7754: get_cluster_log() can fail if 'last_contact'...		Resolved	03/17/2014

Associated revisions

Revision 76753a45 - 04/01/2014 08:29 PM - Dan Mick

conf/salt.master.conf: Allow client to run log_tail

log_tail is run as a LocalClient instance from the Django app, and as such needs permission to run Salt commands as the web user.
Note: both www-user and apache added to keep this file distro-neutral.

Fixes: #7691

Signed-off-by: Dan Mick <dan.mick@inktank.com>

History

#1 - 03/12/2014 03:20 AM - John Spray

This'll probably be because it's running as apache user. We should be able to persuade salt to allow it.

#2 - 03/17/2014 01:13 PM - Dan Mick

Actually, this is now failing in a different way, apparently:

```
File "/opt/calamari/venv/lib/python2.7/site-packages/dateutil/parser.py", line 72, in get_token
    nextchar = self.instream.read(1)
AttributeError: 'NoneType' object has no attribute 'read'
```

which springs from a failure of server_list_cluster(), where servers is apparently None, and thus the lambda in the sorted() call fails. I mention it to wonder if there's a better strategy for handling failed requests in general that should be applied here (realizing that you can't handle every possible error).

#3 - 03/17/2014 08:13 PM - Dan Mick

Ah. This is actually a different bug (in that some servers have "last_contact" == None). I'll file that separately. Once fixed with a custom cmp() function, we return to the EAuthAuthenticationError.

#4 - 03/17/2014 09:38 PM - Dan Mick

Following <http://docs.saltstack.com/ref/clientacl.html>, I added this to /etc/salt/master:

```
client_acl:
  www-data:
    - .*
```

and chmod o+w /var/log/salt, and the log started working.

#5 - 03/18/2014 04:41 AM - John Spray

Ah, good to know the client ACL stuff works. We can make this nice and specific to only let www-user run log_tail.*.

It's annoying that LocalClient is trying to write /var/log/salt, but kind of makes sense because we're passing it just the same configuration as the main salt daemons. I think its worth making it use the main calamari_web log instead, as it'll save us from yet another magic dir permission and is where one would expect the log output anyway.

Could be as simple as something like doing `client.opts['log_file'] = (config.get('calamari_web', 'log_path'))` immediately after constructing LocalClient (haven't tried it though).

#6 - 03/18/2014 04:46 AM - John Spray

- *Tracker changed from Bug to Fix*

#7 - 03/19/2014 11:01 AM - Dan Mick

- *Status changed from New to In Progress*

- *Assignee set to Dan Mick*

#8 - 03/19/2014 11:01 AM - Dan Mick

- *Target version set to v1.2-dev6*

#9 - 03/25/2014 03:07 PM - Dan Mick

Limiting the ACL scope makes sense and is easy. I can't figure out a way to change the logging configuration; I tried setting the field as John suggested, but it didn't take effect; so I tried setting it in the instantiation of LocalClient (in the `mopt=` dict, which failed because it needs the whole config, and then in the a log in the `mopt=` dict param to `salt.client.LocalClient`, but it needs the whole config, so then tried creating a dict with `salt.config.client_config()` as `LocalClient.__init__()` does, and modifying 'log_file', and passing that as `mopt`, but that also failed. I imagine I could go screw with the handlers directly, but that seems painful and fragile.

#10 - 03/27/2014 05:59 AM - John Spray

So I may have misunderstood -- was the log thing actually an obstacle? When I set the `client_acl`, things "just worked", I think salt is just ignoring logging if it doesn't have access to its desired path, or possibly never doing any logging to begin with if we haven't called the `setup_*` functions for logging?

#11 - 04/01/2014 01:31 PM - Dan Mick

- *Status changed from In Progress to Fix Under Review*

#12 - 04/01/2014 01:32 PM - Ian Colle

- *Target version changed from v1.2-dev6 to v1.2-dev7*

#13 - 04/15/2014 01:04 PM - Dan Mick

- *Status changed from Fix Under Review to Resolved*