

Arvados - Bug #6833

[FUSE] arv-mount caches manifests too long, ends up using expired Keep signatures

07/30/2015 09:39 PM - Tom Clegg

Status: Resolved	Start date: 07/30/2015
Priority: Normal	Due date:
Assigned To: Peter Amstutz	% Done: 100%
Category: FUSE	Estimated time: 0.00 hour
Target version: 2016-02-03 Sprint	
Description	
Caught a VM getting 401 like this: 2015-07-30_21:20:00.75039 KeepReadError: failed to read 2e070b0bd2d5c3ca62ac8b63e7fb7d17+66595+A73d20470a444372d5120f3bc8a8e25c846576408@55b01cd1: service http://keep7.su921.arvadosapi.com:25107/ responded with 401 HTTP/1.1 401 Unauthorized	
That block signature expiry time ("55b01cd1") is ~7 days earlier than the log message. Same request worked fine after restarting arv-mount.	
A general solution would be ideal, but I think it should be easy to make arv-mount handle this common case: when checking for a cached copy of a manifest, it should check the expiry times. If any are in the past (or in the very near future) it should replaced the cached manifest with a new one retrieved from the API server.	
Subtasks:	
Task # 8279: Review 6833-test-token-expiry	Resolved
Task # 8276: Find out whether 6833-arv-mount-cache-refresh makes the test in 6833-test-...	Resolved
Task # 8127: Review branch: 6833-arv-mount-cache-refresh	Resolved
Related issues:	
Related to Arvados - Bug #6771: Getting 'Input/output error' on su92l keep mount	New 07/24/2015
Related to Arvados - Bug #10008: [SDKs] [Python] When reading data through Co...	New 09/12/2016
Has duplicate Arvados - Bug #7866: arv-mount not loading all data on qr1hi	Duplicate 11/25/2015

Associated revisions

Revision 97fe5c77 - 01/20/2016 09:15 PM - Radhika Chippada

refs #6833
Merge branch '6833-arv-mount-cache-refresh'

Revision 1034616d - 01/21/2016 10:25 PM - Peter Amstutz

Merge branch '6833-test-token-expiry' closes #6833

Revision 13ca6c96 - 01/22/2016 01:23 PM - Peter Amstutz

Fix python sdk tests refs #6833

Revision 61124439 - 04/19/2016 08:57 PM - Tom Clegg

6833: Fix excessive debug logging in TokenExpiryTest and subsequent tests.

Excessive logging was introduced (seemingly unintentionally) in d3313e65.

refs #6833

History

#1 - 12/01/2015 07:31 PM - Brett Smith

- Story points changed from 0.5 to 2.0

#2 - 01/06/2016 05:08 PM - Brett Smith

- Assigned To set to Radhika Chippada

- Target version changed from Arvados Future Sprints to 2016-01-20 Sprint

#3 - 01/07/2016 04:43 PM - Peter Amstutz

I think this can be as simple as, in `CollectionDirectory.__init__`:

```
self.poll = True
self.poll_time = 60 * 60
```

This should cause it to refresh at minimum every hour.

A slightly more clever solution would be to look up the expiry time in the discovery document and divide by half.

#4 - 01/07/2016 06:52 PM - Radhika Chippada

- Status changed from New to In Progress

#5 - 01/08/2016 02:06 PM - Brett Smith

Peter Amstutz wrote:

I think this can be as simple as, in `CollectionDirectory.__init__`:

```
self.poll = True
self.poll_time = 60 * 60
```

This basic approach seems sound, but this will still have the original bug if the TTL is an hour or less, which I could imagine happening in very security-sensitive deployments that want to be able to revoke access quickly. Could we do something like `self.poll_time = ttl_from_discovery_document / 2`, with some bounds checking and all that?

#6 - 01/11/2016 02:39 PM - Radhika Chippada

Branch 6833-arv-mount-cache-refresh retrieves `blobSignatureTtl` from discovery document and uses it to set `self.poll_time`.

#7 - 01/15/2016 09:31 PM - Tom Clegg

6833-test-token-expiry [ef5692e0](#) has a failing test case for this. If you fill in this part:

- ```
██████████ # TODO: mock something in arvados_fuse here so it thinks
 # manifests/signatures expire in 1 second
```

...then it should reproduce the bug: i.e., the manifest is fetched, then time passes, then fuse attempts to read data using the old signatures.

A more realistic test would be to mock the collections API so the signatures really *do* expire in 1 second. However, since the proposed fix involves looking at the TTL advertised in the discovery doc and ignoring the signatures themselves, mocking the signatures wouldn't be enough to trigger the fix, so we might as well just mock the discovery doc and ensure fuse retrieves fresh (now+2w) signatures from the API server.

### #8 - 01/18/2016 05:06 PM - Radhika Chippada

I am having an issue using the strategy from [ef5692e0](#) in writing a test for this update.

The issue is: the test uses `local_store` and kernel caching for the fuse mount. Due to this I am not able to write a test that does something like: `listdir`, `read file`; `sleep ttl`; `read file` where token is expired.

The other alternative is to write a test with steps: `listdir`, `sleep ttl`, `read file` where token is expired. However, in [ef5692e0](#), the `__test_refresh_old_manifest` method is being executed even before any other test steps are executed (may be because it is a `@staticmethod`). It would help if I can figure out how to make this fire when `get` is actually invoked, rather than at initialization of test.

### #9 - 01/20/2016 08:59 PM - Peter Amstutz

- Target version changed from 2016-01-20 Sprint to 2016-02-03 Sprint

### #10 - 01/20/2016 09:00 PM - Peter Amstutz

- Assigned To changed from Radhika Chippada to Peter Amstutz

### #11 - 01/20/2016 09:06 PM - Peter Amstutz

- Story points changed from 2.0 to 0.5

### #12 - 01/21/2016 08:28 PM - Tom Clegg

Reviewing 6833-test-token-expiry @ [8a2645e](#) (the parts I didn't write)

Please rebase this to master, and mock the discovery document TTL instead of the `poll_time` attribute of the mount. As written, the test passes even though the bug is still present. For example, I'm noticing the bugfix says things about `self.poll`, not `self._poll`, and `ProjectDirectory`'s constructor looks

like it will overwrite whatever CollectionDirectory puts in self.\_poll.

**#13 - 01/21/2016 10:21 PM - Tom Clegg**

[f82d809](#) LGTM

Re MountTestBase... the main improvements in IntegrationTest over MountTestBase are

- it uses the real fuse setup code, so you test the way the mounts get set up in real life, which is a bit hairy
- you can make a class with a bunch of tests (one test\_\* and one matching static test\* method) instead of having to make a class and a lone function outside the class for each test

It also uses a non-admin user by default -- right now the tests are pretty heavy on the "how well things work for admins" side.

**#14 - 01/21/2016 10:30 PM - Peter Amstutz**

- *Status changed from In Progress to Resolved*

- *% Done changed from 67 to 100*

Applied in changeset [1034616dcc47472072a5c6e2d7b92c9b95c544c9](#).