Arvados - Bug #5724

[API] Advertise BlobSignatureTTL in discovery doc. Fix name/comments on default_trash_lifetime.

04/14/2015 05:36 PM - Tom Clegg

Status:	Resolved	**Start date:**	04/30/2015
Priority:	Normal	**Due date:**	
Assigned To:	Tom Clegg	**% Done:**	100%
Category:	API	**Estimated time:**	0.00 hour
Target version:	2015-05-20 sprint		

Description

The default_trash_lifetime config setting (and the defaultTrashLifetime discovery doc key) are currently unused and were intended to control the default interval after pushing the "trash" button on an object in Workbench (or doing a similar action) during which the object could be rescued/undeleted. This is achieved by setting expires_at=now()+defaultTrashLifetime on an object. Note:

- This is not the same kind of "trash" as unreferenced data blocks in Keep.
- This is only a default: a client can also choose to set expires_at to now()+1h or now()+8y.
- A larger value makes it more likely a user can recover accidentally deleted data.
- A smaller value conserves disk space.
- This is not a security feature.

Meanwhile, the API server's blob_signing_ttl config setting refers to the permission signature hints on Keep locators. When the API server sends manifest_text to a client, it includes signature hints that claim to expire (and do in fact expire) at now()+blob_signing_ttl. Clients expect to be able to read the blocks until that time, without any further contact with the API server. Note:

- Whether the collection's expires_at is now+1h, now+8y, or null, the signature TTL is the same (default is 2 weeks).
- This is a security feature. It prevents a client from stockpiling signatures over time and reading data which it had permission to read 8 years ago (but doesn't now).

Closely related is keepstore's -blob-signing-ttl command line flag. This is expected to be equal to the API server's blob_signing_ttl config setting, but currently no mechanism is in place to guarantee this.

- This limits the time a client can take to build a collection.
- Like the API server's ttl, this TTL is visible to clients (the expiry time part is easy to parse out of the blob signature) so clients can be expected to make decisions on this basis ("the block signature will expire soon, so I'd better create a collection to get a fresh one"). Therefore, a keepstore that has provided a signature expiring on January 15 must not delete the corresponding data before January 15 merely because that block hasn't appeared in any collections yet.
- keepstore should (but currently doesn't) advertise its blob_signing_ttl to clients explicitly. Currently, clients can figure it out by writing a block and subtracting now() from the resulting signature expiry time.

Subtasks:

Task # 5880: Review 5724-blob-signature-ttl	**Resolved**
Task # 5879: Review 5724-blob-signature-ttl at 6fc44a6	**Resolved**
Task # 5859: Update default config, add to discovery doc	**Resolved**

Associated revisions

Revision eb03fb3a - 05/04/2015 02:57 PM - Tom Clegg

Merge branch '5724-blob-signature-ttl' commit '6fc44a6' refs #5724

Revision 9d4bc458 - 05/04/2015 08:46 PM - Tom Clegg

Merge branch '5724-blob-signature-ttl' closes #5724

History

#1 - 04/14/2015 05:58 PM - Tom Clegg

- Description updated

- Category set to API

#2 - 04/29/2015 07:06 PM - Ward Vandewege

- Target version changed from 2015-04-29 sprint to 2015-05-20 sprint

#3 - 04/29/2015 07:22 PM - Tom Clegg

- Assigned To set to Tom Clegg

#4 - 04/30/2015 08:52 PM - Tom Clegg

- Status changed from New to In Progress

#5 - 05/04/2015 02:17 PM - Radhika Chippada

Comments for branch 5724-blob-signature-ttl at 6fc44a6:

Should the comment at line 234 in application.default.yml be using "blob_signature_ttl" instead of "blob-signing-ttl"? (It appears that you did not update it in newer version also; so please update it in the newer code if you want to do this with other keepstore related updates.)

Everything else lgtm.

#6 - 05/04/2015 04:25 PM - Radhika Chippada

Comments with the latest in branch (da57a997):

- Since you are renaming variables, I was wondering if we want to call "*signing-key*" as "*signature-key*"?

- One test is failing for me in keepstore. Could be my environment; if it is passing for you please ignore and I will try to resolve my env isssues.

```
2015/05/04 12:18:33 [b1f792e1] Upload failed http://localhost:40948/5d41402abc4b2a76b9719d911017c592 error: Pu
t http://localhost:40948/5d41402abc4b2a76b9719d911017c592: dial tcp 127.0.0.1:40948: connection refused
2015/05/04 12:18:33 DEBUG: GET 5d41402abc4b2a76b9719d911017c592 failed: [http://localhost:59768/5d41402abc4b2a
76b9719d911017c592: Get http://localhost:59768/5d41402abc4b2a76b9719d911017c592: dial tcp 127.0.0.1:59768: con
nection refused http://localhost:40948/5d41402abc4b2a76b9719d911017c592: Get http://localhost:40948/5d41402abc
4b2a76b9719d911017c592: dial tcp 127.0.0.1:40948: connection refused]
--- FAIL: TestPullWorkerIntegration_GetExistingLocator (7.72 seconds)
    pull_worker_integration_test.go:56: Error putting test data in setup for hello  Could not write sufficient
 replicas
    pull_worker_integration_test.go:59: No locator found after putting test data
    pull_worker_integration_test.go:123: Got error Block not found
2015/05/04 12:18:33 Pull {locator3 [server_1 server_2]} error: Error getting data
```

- If the tests are passing for you, lgtm.

#7 - 05/04/2015 08:21 PM - Tom Clegg

Radhika Chippada wrote:

> Comments with the latest in branch (da57a997):
>
> - Since you are renaming variables, I was wondering if we want to call "*signing-key*" as "*signature-key*"?

I thought about that too, but "signature key" sounds a bit odd to me where "signing key" and "signature ttl" sound natural. I think the difference is that the TTL is an attribute of each signature -- you could say the key is an *input* to the signature, mathematically, but it's not an observable attribute in the way the TTL is. I do like the consistency of naming all of the related bits "blob_signature_something", though.

> - One test is failing for me in keepstore. Could be my environment; if it is passing for you please ignore and I will try to resolve my env isssues.
>
> [...]
>
> - If the tests are passing for you, lgtm.

Haven't seen that one. I'm guessing this is a race condition similar to #5898, where the server child is running but hasn't listened on the port yet. We should probably have a wait loop in run_test_servers.py→_start_keep() so it waits up to X milliseconds for the port to listen (by reading /proc/net/tcp?) and returns or raises depending on the outcome.

#8 - 05/04/2015 09:05 PM - Tom Clegg

- Status changed from In Progress to Resolved

- % Done changed from 47 to 100

Applied in changeset arvados|commit:9d4bc458b767e4c05024dfe02207283745e1ba06.