Arvados - Bug #16736

Token lifetime options

08/20/2020 04:19 PM - Lucas Di Pentima

Status:	Resolved	**Start date:**	09/10/2020
Priority:	Normal	**Due date:**	
Assigned To:	Lucas Di Pentima	**% Done:**	100%
Category:	API	**Estimated time:**	0.00 hour
Target version:	2021-03-03 sprint		

Description

Add a new option API.MaximumTokenLifetime:

- If no expiration time is given in the create call, it is the "maximum" expiration (new configuration option API.MaximumTokenLifetime)
- For regular users, token expires_at is clamped to MaximumTokenLifetime for create/update
- Admins can create tokens with any expiration time.
- Tokens created to run a container do not have a set expire time (because it will expire when the container ends)
- Tokens created for use on a shell node by arvados-login-sync script have max lifetime, and are rotated by the script on some interval (like MaximumTokenLifetime/2)

Tokens created through login use Login.TokenLifetime (existing behavior).

Subtasks:

Task # 16754: Review 16736-expiring-tokens-limits	**Closed**
Task # 17034: Review 16736-max-token-lifetime	**Resolved**

Related issues:

Related to Arvados Epics - Story #16520: GxP Qualification	**Resolved**	**08/01/2020**	**04/30/2021**

Associated revisions

Revision b3e4886c - 02/19/2021 08:30 PM - Lucas Di Pentima

Merge branch '16736-max-token-lifetime'
Closes #16736

Arvados-DCO-1.1-Signed-off-by: Lucas Di Pentima <lucas@di-pentima.com.ar>

History

#1 - 08/20/2020 04:48 PM - Lucas Di Pentima

Pushed a branch with a test: 16736-expiring-tokens-limits

#2 - 08/24/2020 08:31 PM - Lucas Di Pentima

- Status changed from New to In Progress

#3 - 08/24/2020 08:32 PM - Lucas Di Pentima

- Target version changed from 2020-09-09 Sprint to 2020-08-26 Sprint

#4 - 08/26/2020 03:32 PM - Lucas Di Pentima

- Target version changed from 2020-08-26 Sprint to 2020-09-09 Sprint

#5 - 09/09/2020 02:16 PM - Lucas Di Pentima

- Target version changed from 2020-09-09 Sprint to 2020-09-23 Sprint

#6 - 09/10/2020 03:29 PM - Lucas Di Pentima

Updates at 386b60af1 - branch 16736-expiring-tokens-limits
Test run: https://ci.arvados.org/job/developer-run-tests/2078/

- Adds tests exposing the bug.
- Adds checks related to token's expires_at on create/update.
- Fixes fixture removing expiration dates that were set far in the future.

#7 - 09/10/2020 06:40 PM - Peter Amstutz

Lucas Di Pentima wrote:

> Updates at 386b60af1 - branch 16736-expiring-tokens-limits
> Test run: https://ci.arvados.org/job/developer-run-tests/2078/
>
> - Adds tests exposing the bug.
> - Adds checks related to token's expires_at on create/update.
> - Fixes fixture removing expiration dates that were set far in the future.

```
  def permission_to_create
    current_user.andand.is_admin or
      ((current_user.andand.id == self.user_id) and
        (current_api_client_authorization.andand.expires_at.nil? or
          (self.expires_at and current_api_client_authorization.expires_at >= self.expires_at)))
  end
```

I don't think this is quite right, this will allow an admin to always create a token even if their token is about to expire. But that's not right. Admins can create tokens for other users, but the tokens are still capped by expires_at.

Also, instead of "current_api_client_authorization.andand", if it is possible for current_api_client_authorization to be nil I don't think we allow tokens to be manipulated (is there a special case, like tests?)

I think this should be:

```
  def permission_to_create
    (current_user.andand.is_admin ||
      current_user.andand.id == self.user_id) and
        !current_api_client_authorization.nil? and
        (current_api_client_authorization.expires_at.nil? ||
          (!self.expires_at.nil? && current_api_client_authorization.expires_at >= self.expires_at)))
  end
```

By the way, I looked this up recently, the difference between "and"/"or" and "&&"/"||" in Ruby has to do with operator precedence, "and"/"or" have lower precedence than "&&"/"||" so the convention is to use the and/or for the "outer" clauses and &&/|| for the "inner" clauses although when grouping with parenthesis it doesn't really matter.

Arvados::V1::ApiClientAuthorizationsController#find_objects_for_index should include a filter on expires_at when current_api_client_authorization is non-nil, and there should be a controller test for that.

#8 - 09/14/2020 08:13 PM - Peter Amstutz

update: I think you mentioned on gitter than current_api_client_authorization is nil in the case of the login process, which makes sense. We are creating a token, we don't have one yet.

However, if there was a bug that allowed some other way of calling token creation without setting current_api_client_authorization, that would be a serious security hole.

I wonder if there's a way to handle login as a special case. The value of current_api_client_authorization comes from Thread.current[:api_client_authorization], so the login code could assign a fake (not persisted to database) ApiClientAuthorization object.

#9 - 09/17/2020 08:01 PM - Peter Amstutz

When Login.TokenLifetime is set:

- Token from login uses "Login.TokenLifetime" expiration (existing behavior)
- (valid) token from login can be used to create new tokens
  - If no expiration time is given, it is the "maximum" expiration (new configuration option API.MaximumTokenLifetime)
  - SystemRootToken can create tokens with any expiration (or nil)
- Token created to run a container doesn't have a set expire time, because it will expire when the container ends
- Token used on a shell node is max lifetime, when about to expire it is rotated by the arvados-login-sync script
- "Show current token" dialog in workbench should generate a new token instead
  - Change the label to something like "Get new user token"
- Auto-logout should coordinate across browser tabs/windows (using the same token?) so that it only logs out when all of them have been idle.
  - Figure out which browser feature lets you do that

#10 - 09/17/2020 08:22 PM - Peter Amstutz

- Release deleted (25)

#11 - 09/17/2020 09:28 PM - Peter Amstutz

- Subject changed from Trusted client token with expiration can create tokens with any expiration date (or none at all) to Token default and maximum lifetime option

#12 - 09/17/2020 10:02 PM - Peter Amstutz

- Target version changed from 2020-09-23 Sprint to 2020-10-07 Sprint

- Assigned To deleted (Lucas Di Pentima)

- Description updated

- Subject changed from Token default and maximum lifetime option to Token lifetime options

#13 - 09/23/2020 04:15 PM - Peter Amstutz

- Target version changed from 2020-10-07 Sprint to 2020-10-21 Sprint

#14 - 10/07/2020 04:09 PM - Peter Amstutz

- Target version changed from 2020-10-21 Sprint to 2020-11-04 Sprint

#15 - 10/21/2020 02:08 PM - Peter Amstutz

- Related to Story #16520: GxP Qualification added

#16 - 10/21/2020 04:30 PM - Peter Amstutz

- Assigned To set to Peter Amstutz

#17 - 11/03/2020 08:04 PM - Peter Amstutz

- Target version changed from 2020-11-04 Sprint to 2020-11-18

#18 - 11/16/2020 06:42 PM - Peter Amstutz

- Target version changed from 2020-11-18 to 2020-12-02 Sprint

#19 - 11/17/2020 06:26 PM - Peter Amstutz

- Story points set to 3.0

#20 - 11/18/2020 04:45 PM - Peter Amstutz

- Assigned To deleted (Peter Amstutz)

#21 - 11/18/2020 04:47 PM - Peter Amstutz

- Target version changed from 2020-12-02 Sprint to 2020-12-16 Sprint

#22 - 12/02/2020 04:53 PM - Peter Amstutz

- Assigned To set to Lucas Di Pentima

#23 - 12/16/2020 02:43 PM - Lucas Di Pentima

- Target version changed from 2020-12-16 Sprint to 2021-01-06 Sprint

#24 - 01/06/2021 04:54 PM - Lucas Di Pentima

- Target version changed from 2021-01-06 Sprint to 2021-01-20 Sprint

#25 - 01/20/2021 03:18 PM - Lucas Di Pentima

- Target version changed from 2021-01-20 Sprint to 2021-02-03 Sprint

#26 - 02/03/2021 04:34 PM - Lucas Di Pentima

- Target version changed from 2021-02-03 Sprint to 2021-02-17 sprint

#27 - 02/04/2021 07:04 PM - Lucas Di Pentima

Updates at bb5d1bb61 - branch 16736-max-token-lifetime
Test run: https://ci.arvados.org/job/developer-run-tests/2301/

- Adds API.MaxTokenLifetime configuration.
- Limits token's expires_at when the configuration is set and users don't provide a valid expiration date on creation or update.
- Adds options to arvados-login-sync to allow the admin to set it up in a way that periodically rotates expiring tokens on shell sessions.

#28 - 02/05/2021 10:06 PM - Peter Amstutz

- clamp_token_expiration:

```
max_token_expiration = Time.now + Rails.configuration.API.MaxTokenLifetime
```

I think we want to be using db_current_time here?

Related, I see another use of Time.now in the validate method which I believe should also be db_current_time

- I'm not sure I see the purpose of elsif here:

```
      if self.new_record? && (self.expires_at.nil? || self.expires_at > max_token_expiration)
        self.expires_at = max_token_expiration
      elsif !self.new_record? && self.expires_at_changed? && (self.expires_at.nil? || self.expires_at > max_token_expiration)
        self.expires_at = max_token_expiration
      end
```

Would this do the same thing?

```
if (self.new_record? || self.expires_at_changed?) && (self.expires_at.nil? || self.expires_at > max_token_expiration)
   self.expires_at = max_token_expiration
end
```

- Did you confirm that tokens created for containers do not have an expiration set? Is there a test to check that? (it is unclear if the container assign_auth method is guaranteed to be called by the system user).

#29 - 02/12/2021 12:11 AM - Lucas Di Pentima

Updates at 46a5e39 (rebased) - branch 16736-max-token-lifetime
Test run: https://ci.arvados.org/job/developer-run-tests/2327/

Peter Amstutz wrote:

> I think we want to be using db_current_time here?
> Related, I see another use of Time.now in the validate method which I believe should also be db_current_time

Yes, thank you, fixed!

> - I'm not sure I see the purpose of elsif here:
>   Would this do the same thing?

Fixed too.

> - Did you confirm that tokens created for containers do not have an expiration set? Is there a test to check that? (it is unclear if the container assign_auth method is guaranteed to be called by the system user).

As I mentioned on today's standup: the assign_auth case doesn't get a clamped expires_at because that code is run with dispatcher credentials when the container is locked (added comment for clarification).
In the case of a federated call, the controller creates the runtime token so it also doesn't get clamped.
I added a test for the assign_auth case and expanded a preexisting one for the federated call case.

#30 - 02/18/2021 04:35 PM - Lucas Di Pentima

- Target version changed from 2021-02-17 sprint to 2021-03-03 sprint

#31 - 02/19/2021 08:28 PM - Peter Amstutz

Lucas Di Pentima wrote:

> Updates at 46a5e39 (rebased) - branch 16736-max-token-lifetime
> Test run: https://ci.arvados.org/job/developer-run-tests/2327/
>
> Peter Amstutz wrote:
>
>> I think we want to be using db_current_time here?
>> Related, I see another use of Time.now in the validate method which I believe should also be db_current_time
>
> Yes, thank you, fixed!

- I'm not sure I see the purpose of elsif here:
  Would this do the same thing?

Fixed too.

- Did you confirm that tokens created for containers do not have an expiration set? Is there a test to check that? (it is unclear if the container assign_auth method is guaranteed to be called by the system user).

As I mentioned on today's standup: the assign_auth case doesn't get a clamped expires_at because that code is run with dispatcher credentials when the container is locked (added comment for clarification).
In the case of a federated call, the controller creates the runtime token so it also doesn't get clamped.
I added a test for the assign_auth case and expanded a preexisting one for the federated call case.

This LGTM, thanks.

#32 - 02/19/2021 10:01 PM - Anonymous

- Status changed from In Progress to Resolved

Applied in changeset arvados|b3e4886cbbe195347179d0664621da9bc34e6170.

#33 - 11/16/2021 04:46 PM - Peter Amstutz

- Release set to 41