Pulp - Story #8939

Add token authentication to pulpcore

06/21/2021 05:57 PM - gerrod

Status:	ASSIGNED	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	gerrod	**% Done:**	0%
Category:		**Estimated time:**	0:00 hour
Sprint/Milestone:			
Platform Release:		**Tags:**	GalaxyNG
Groomed:	No	**Sprint:**	
Sprint Candidate:	No	**Quarter:**	

Description

Background

Token authentication hands out a secret token to be used by a user to authenticate themselves. These tokens are passed in through the Authorization HTTP header with each request usually in the form of TOKEN {USER_TOKEN}. Tokens need to be kept secret like passwords and should only be used with https. Token auth can be implemented many different ways, but the general workflow follows:

1. User visits a token view with basic credentials to receive a token
2. Server generates a token that can be used to authenticate that user for future requests
3. User uses token for token for authentication on future requests
4. Token expires after set time or user deletes/generates a new token

Token auth can easily be added using the pre-built token authentication available in DRF: https://www.django-rest-framework.org/api-guide/authentication/#tokenauthentication. Two popular methods for token authentication are simple HTTP tokens and JSON Web token (JWT).

Simple HTTP Tokens

Basic token authentication comes included in DRF and can be added by including rest_framework.authtoken in the INSTALLED_APPS list and by adding TokenAuthentication to DRF's DEFAULT_AUTHENTICATION_CLASSES setting. This creates a model for storing tokens in the database. obtain_auth_token is a default view for generating tokens that can be added for users to receive their tokens.

Pros

- Simple to add and customize
- Admins and users can track tokens that have been deployed
- Lots of third party libraries that are built on top to add more functionality: Django-Rest-Durin, drfpasswordless, django-rest-knox, Djoser

Cons

- Adds an extra model to the database to maintain.

JSON Web Tokens

Background: https://jwt.io/introduction. JWTs consist of three encoded strings separated by dots which are: the header, the payload and the signature. JWTs use the signatures of the token to validate the authenticity of a token and have no need for a database to store them. The header tells how the token is encoded and the payload contains information, called claims, about the token like the user and expiration time for the token. The signature is created from the private key of the server and the encoded strings of the header and payload to validate the token. JWTs usually use the Bearer schema inside the Authentication header, e.g.: Authentication: Bearer {USER_TOKEN}. Since the signature validates the token an expiration is added to the payload to be able to invalidate it. djangorestframework-simplejwt implements JWTs for DRF by generating two tokens for users. One is a short-lived access token to authenticate and the other is a longer-lived refresh token that can be used to get another acess token.

Pros

- Simple to add using djangorestframework-simplejwt
- No need for database models
- Tokens expiration naturally built in

Cons

- No way to manually delete/track tokens without a database table
- Tokens need to be continuously fetch since long-lived tokens are ill-advised

Implementation Options

1. Just use DRF's token auth or a pre-built DRF token auth package
2. Just use JWT from djangorestframework-simplejwt
3. Build a custom implementation on top of DRF tokens

Related issues:

Related to Ansible Plugin - Story #7118: As an ansible-galaxy CLI user, I can...	**CLOSED - DUPLICATE**

History

#1 - 06/21/2021 05:58 PM - gerrod

- Related to Story #7118: As an ansible-galaxy CLI user, I can configure a token and auth_url and have pulp_ansible protect my content added

#2 - 06/21/2021 08:12 PM - daviddavis

- Sprint/Milestone set to 3.15.0

#3 - 06/22/2021 12:13 PM - ipanova@redhat.com

Has it been decided which implementation option will it be? In case jwt tokens, I'd like to see code ported from pulp-container into core so more plugins can benefit from it.

#4 - 07/20/2021 04:29 PM - daviddavis

- Tags GalaxyNG added

#5 - 07/21/2021 03:54 PM - gerrod

- Status changed from NEW to ASSIGNED

#6 - 08/05/2021 05:53 PM - dkliban@redhat.com

- Sprint/Milestone deleted (3.15.0)