Arvados - Feature #16669

Accept OpenID Connect access token

08/06/2020 05:25 PM - Peter Amstutz

Status:	Resolved	**Start date:**	09/24/2020
Priority:	Normal	**Due date:**	
Assigned To:	Tom Clegg	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2021-03-17 sprint		

Description

When getting an unrecognized token, add an option to validate the token against an OpenID Connect provider.

1. Determine if the token is valid & when it expires using the OAuth2 token Introspection endpoint https://tools.ietf.org/html/rfc7662
2. If valid and not expired, make a call to the UserInfo endpoint of the provider, this will return similar claims as the existing log in process, or an error. https://openid.net/specs/openid-connect-core-1_0.html#UserInfo
3. Cache the token in the Arvados database along with the expiration time.

If a LoginCluster is configured, the token is checked with the upstream LoginCluster (only change is that this happens for JWT tokens and not just v2 tokens).

The endpoint URLs to the Introspection and UserInfo endpoints can be discovered by looking at the "provider configuration" endpoint.

https://openid.net/specs/openid-connect-discovery-1_0.html

https://docs.pingidentity.com/bundle/pingfederate-101/page/bwm1564003025542.html

Additional notes:

Accepting OpenID access tokens

Subtasks:

Task # 16757: Review 16669-oidc-access-token	**Resolved**
Task # 17023: Manual testing federation case on dev clusters	**Resolved**
Task # 17294: Review 16669-oidc-access-token	**Resolved**
Task # 17453: Review 16669-oidc-access-token-fed	**Resolved**

Related issues:

Related to Arvados - Feature #17037: [controller] Improve use of given_name/f...	**New**		
Related to Arvados - Feature #17038: [controller] Option to request additiona...	**New**		
Related to Arvados - Bug #16774: keep-web needs to return user-visible errors	**Resolved**	**11/20/2020**	
Related to Arvados Epics - Story #16360: Keep-web supports S3 compatible inte...	**Resolved**	**07/01/2020**	**04/30/2021**
Related to Arvados - Feature #17468: [controller] Skip repetitive OIDC UserIn...	**New**		

Associated revisions

Revision dfeee281 - 10/21/2020 02:11 PM - Tom Clegg

Merge branch '16669-oidc-access-token'

refs #16669

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tom@tomclegg.ca>

Revision 6470f7ce - 01/26/2021 08:44 PM - Tom Clegg

Merge branch '16669-oidc-access-token'

refs #16669

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tom@curii.com>

Revision f737eafb - 03/10/2021 07:22 PM - Tom Clegg

Merge branch '16669-oidc-access-token-fed'

refs #16669

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tom@curii.com>

History

#1 - 08/06/2020 06:02 PM - Peter Amstutz

- Description updated

#2 - 08/06/2020 06:23 PM - Peter Amstutz

- Description updated

#3 - 08/06/2020 06:24 PM - Peter Amstutz

- Description updated

#4 - 08/26/2020 04:19 PM - Peter Amstutz

- Assigned To set to Tom Clegg

#5 - 08/26/2020 08:24 PM - Tom Clegg

- Status changed from New to In Progress

#6 - 08/27/2020 03:45 PM - Tom Clegg

- Description updated

#7 - 09/09/2020 03:02 PM - Tom Clegg

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

#8 - 09/16/2020 03:09 PM - Tom Clegg

16669-oidc-access-token @ 61d493d84cff707e291132847ac3e9792ae94aee -- developer-run-tests: #2096 icon?job=developer-run-tests&amp;build=2096

- We don't get an expiry time from the access token. It's not clear (to me) whether the introspection API is likely to be supported/available, and the Go library doesn't provide support for it, so for now this code always treats the OIDC access token expiry as "unknown" and re-verifies after 5m.

#9 - 09/22/2020 05:32 PM - Peter Amstutz

Tom Clegg wrote:

> 16669-oidc-access-token @ 61d493d84cff707e291132847ac3e9792ae94aee -- developer-run-tests: #2096 icon?job=developer-run-tests&amp;build=2096
>
> - We don't get an expiry time from the access token. It's not clear (to me) whether the introspection API is likely to be supported/available, and the Go library doesn't provide support for it, so for now this code always treats the OIDC access token expiry as "unknown" and re-verifies after 5m.

PingFederate supports the introspection API, so ideally we should support it if available.

#10 - 09/22/2020 07:09 PM - Tom Clegg

16669-oidc-access-token @ aa76fee75c0e96214d2ffc5ecbb1c8a06f70b309 -- developer-run-tests: #2112 icon?job=developer-run-tests&amp;build=2112

#11 - 09/23/2020 03:54 PM - Tom Clegg

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

#12 - 09/24/2020 02:08 PM - Tom Clegg

16669-oidc-access-token @ 43a3273c3aaed1d28c21a83890810eb62783cf32 -- developer-run-tests: #2117 icon?job=developer-run-tests&amp;build=2117

- If LoginCluster is in use, when RailsAPI receives a non-v2 token that isn't [unexpired] in the local database, it asks LoginCluster's controller to verify it.
- If OIDC login is in use, when controller receives a non-v2 token that isn't [unexpired] in the local database, it asks the OIDC provider to verify it.
  - When controller accepts an OIDC access token, it is cached using hmac(systemroottoken,oidcaccesstoken) as the secret part (api_token

column) in the local database.
- When controller's federation delegate/fan-out code needs to send a request to a remote cluster and it's working with a non-v2 token, it first checks whether the token belongs to a user account on the target cluster, and if so, it passes it through instead of making a salted token.

#13 - 09/25/2020 06:35 PM - Peter Amstutz

Question: should we only support a login flow that exchanges the OIDC access token for an Arvados token, instead of accepting the access token for every API call?

This is based on:

https://www.ory.sh/hydra/docs/concepts/before-oauth2

which among other things states

> A common misconception is that access and refresh tokens represent a user session.
> ...
> That is because both applications are built by different people and different companies, and the user behavior on one site does not reflect user intent on another. It would be crazy if some app could just log you out of GitHub, or if signing out of Google would invalidate your sessions on all the websites where you used "Sign in with Google".

However, after studying this some more, there's an important distinction: the OAuth2 case described on that page is for "peer" applications where one is granting limited access for one application to interact with another.

But for us, we are supporting the Single-Sign-On case. In that case, it does behave more like a session. When the user logs off from SSO, they should be logged off everywhere.

So I think the objections about using generic OAuth2 for sessions don't apply.

#14 - 09/25/2020 06:52 PM - Peter Amstutz

The result of OIDC login with Google is an access token that can access Google APIs (only the ones with granted scopes, though). So I think that answers the question "are access tokens supposed to be used as API tokens".

#15 - 10/07/2020 01:59 PM - Peter Amstutz

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

#16 - 10/20/2020 09:34 PM - Peter Amstutz

I rebased this branch on master since it was a month old. I force-pushed the rebase.

16669-oidc-access-token @ 64028c0cae469d3da33878a30bd40c5409e64641

Let me see if I understand how this works.

1. We receive a bare token
2. Check local controller cache, which can be a miss, possibly valid (check expiration), or probably invalid (recheck after expiration).
3. Hash the token and check if the hash exists in the database. If it present and still valid, return success.
4. Check that the token is valid from the OIDC provider
5. Create or update an arvados token record based on the information from the OIDC provider, and update the local cache.

Is this comment accurate? I don't think we're swapping anything, we're just ensuring that the Rails API server will accept the token?

> // Check whether the token is a valid OIDC access token. If
> // so, swap it out for an Arvados token (creating/updating an
> // api_client_authorizations row if needed) which downstream
> // server components will accept.

We should have configuration options GivenNameClaim (default "given_name") and FamilyNameClaim (default "family_name") for setting first_name / last_name. Right now we only get those from the Google People API, so the name will be blank for when using another provider.

Standard claims are here:

https://openid.net/specs/openid-connect-core-1_0.html#StandardClaims

I wonder if we should add an option to check/enforce that an OIDC token has been issued with "arvados" scope (or whatever).

For testing: I ran the unit tests and they passed. I'd like to do some manual integration tests, but that is kind of a hassle to set up. Maybe once the other comments above are addressed, we go ahead and merge and then I can follow with some testing on the dev cluster.

#17 - 10/21/2020 01:23 PM - Tom Clegg

> Is this comment accurate? I don't think we're swapping anything, we're just ensuring that the Rails API server will accept the token?

Yes, oidcTokenAuthorizer ensures that if the token is (or was very recently) a valid OIDC access token, then RailsAPI will recognize it (by HMAC) as an Arvados token.

> We should have configuration options GivenNameClaim ...

We could -- does someone need it? That feature wouldn't affect this branch though anyway, right?

> I wonder if we should add an option to check/enforce that an OIDC token has been issued with "arvados" scope (or whatever).

That feature wouldn't affect this branch though, right?

> I'd like to do some manual integration tests

Yes, I've also been looking forward to having this deployed on dev clusters. ;)

16669-oidc-access-token @ b8de8845c856f7fe1232e5f048824211d1207ee7 -- developer-run-tests: #2139 icon?job=developer-run-tests&amp;build=2139

#18 - 10/21/2020 01:46 PM - Peter Amstutz

Tom Clegg wrote:

> > Is this comment accurate? I don't think we're swapping anything, we're just ensuring that the Rails API server will accept the token?
>
> Yes, oidcTokenAuthorizer ensures that if the token is (or was very recently) a valid OIDC access token, then RailsAPI will recognize it (by HMAC) as an Arvados token.
>
> > We should have configuration options GivenNameClaim ...
>
> We could -- does someone need it? That feature wouldn't affect this branch though anyway, right?

Sure, it doesn't have to be a blocker for an initial merge.

I don't know if GivenName needs to be configurable with a GivenNameClaim or if given_name/family_name are actually used consistently across implementations. But we definitely need it because someone using generic OIDC for login wouldn't have correct display names otherwise.

> > I wonder if we should add an option to check/enforce that an OIDC token has been issued with "arvados" scope (or whatever).
>
> That feature wouldn't affect this branch though, right?

Sure.

> > I'd like to do some manual integration tests
>
> Yes, I've also been looking forward to having this deployed on dev clusters. ;)
>
> 16669-oidc-access-token @ b8de8845c856f7fe1232e5f048824211d1207ee7 -- developer-run-tests: #2139 icon?job=developer-run-tests&amp;build=2139

Ok, please merge but let's keep the ticket open for follow-up.

#19 - 10/21/2020 02:19 PM - Peter Amstutz

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

#20 - 10/21/2020 05:34 PM - Tom Clegg

- Related to Feature #17037: [controller] Improve use of given_name/family_name fields for generic OpenID Connect providers added

#21 - 10/21/2020 05:40 PM - Tom Clegg

Peter Amstutz wrote:

> I don't know if GivenName needs to be configurable with a GivenNameClaim or if given_name/family_name are actually used consistently across implementations. But we definitely need it because someone using generic OIDC for login wouldn't have correct display names otherwise.

Added #17037

#22 - 10/21/2020 06:40 PM - Tom Clegg

- Related to Feature #17038: [controller] Option to request additional scopes, and verify additional claims, during OpenID Connect auth added

#23 - 11/04/2020 05:06 PM - Peter Amstutz

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

#24 - 11/12/2020 10:21 PM - Peter Amstutz

s3cmd doesn't like the error that we're returning, I think we're returning a bare string like "signature verification failed: invalid access key" but it is expect some kind of XML structure.

The 404 response has the same issue (a bare string "not found" instead of the XML structure it is expecting).

#25 - 11/13/2020 08:00 PM - Peter Amstutz

This ended up being very helpful in testing: https://github.com/ajanthan/cmdline-openid-client

For the most part, this worked on the first try, which impressive for such a complex feature.

Things that worked:

Got an access token from Google with the "openid profile email" scopes.

1. Exported the access token as ARVADOS_API_TOKEN.
2. Was able to do "arv user current" on ce8i5, and got the correct user.
3. Was able to do "arv user current" on tordo (which has ce8i5 as its LoginCluster). This returned the correct user.
4. Was able to use "s3cmd" to access collections on both ce8i5 and tordo by setting the access token on both access_key/secret_key, and changing host_base to each respective cluster, and requesting a collection hosted on that cluster.

Didn't work:

- Making a federated query, requesting a collection from tordo while connected to ce8i5 via the S3 API. I haven't tried this with a regular Arvados token.
- After 5 minutes, it expired. I'm pretty sure the access token itself is still valid, but it seems like Arvados is not re-checking the token, I'm getting 401 Unauthorized.

#26 - 11/18/2020 04:36 PM - Peter Amstutz

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

#27 - 11/19/2020 04:27 PM - Tom Clegg

- Related to Bug #16774: keep-web needs to return user-visible errors added

#28 - 12/02/2020 04:33 PM - Tom Clegg

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

#29 - 12/02/2020 04:43 PM - Peter Amstutz

- Status changed from In Progress to Feedback

#30 - 12/09/2020 04:46 PM - Peter Amstutz

- Related to Story #16360: Keep-web supports S3 compatible interface added

#31 - 12/16/2020 05:06 PM - Tom Clegg

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

#32 - 01/06/2021 04:59 PM - Peter Amstutz

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

#33 - 01/19/2021 06:46 PM - Peter Amstutz

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

#34 - 01/26/2021 03:11 PM - Tom Clegg

- Status changed from Feedback to In Progress

> After 5 minutes, it expired. I'm pretty sure the access token itself is still valid, but it seems like Arvados is not re-checking the token, I'm getting 401 Unauthorized.

This was a caching bug. Fix+test:

16669-oidc-access-token @ 969441a091ce3aa1eb7a9525d3ab85f24fbd8fdd -- developer-run-tests: #2280 icon?job=developer-run-tests&amp;build=2280

#35 - 01/26/2021 08:10 PM - Lucas Di Pentima

This LGTM.

#36 - 02/03/2021 04:40 PM - Peter Amstutz

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

#37 - 02/18/2021 04:51 PM - Peter Amstutz

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

#38 - 03/03/2021 06:03 PM - Tom Clegg

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

#39 - 03/05/2021 06:02 AM - Tom Clegg

16669-oidc-access-token-fed @ 54d8c4e41a276ac82c79506f63907a108ebd9bfd -- developer-run-tests: #2361 icon?job=developer-run-tests&amp;build=2361

- fix & test for using OIDC tokens in federated calls
- test case has all clusters using the same LoginCluster
- should also work with different LoginClusters, but only if the OIDC providers all recognize one another's access tokens, which practically speaking probably means using the same OIDC Issuer, ~~ClientID, and ClientSecret.~~

#40 - 03/09/2021 04:32 PM - Peter Amstutz

Tom Clegg wrote:

> 16669-oidc-access-token-fed @ 54d8c4e41a276ac82c79506f63907a108ebd9bfd -- developer-run-tests: #2361 icon?job=developer-run-tests&amp;build=2361
>
> - fix & test for using OIDC tokens in federated calls
> - test case has all clusters using the same LoginCluster
> - should also work with different LoginClusters, but only if the OIDC providers all recognize one another's access tokens, which practically speaking probably means using the same OIDC Issuer, ClientID, and ClientSecret.

LGTM

#41 - 03/11/2021 03:36 PM - Tom Clegg

- Related to Feature #17468: [controller] Skip repetitive OIDC UserInfo calls if access token validates as an ID token added

#42 - 03/11/2021 04:35 PM - Tom Clegg

- Release set to 38

- Status changed from In Progress to Resolved