Arvados - Feature #16171

Support generic OpenID Connect login provider

02/25/2020 03:53 PM - Tom Clegg

Status:	Resolved	**Start date:**	06/01/2020
Priority:	Normal	**Due date:**	
Assigned To:	Tom Clegg	**% Done:**	100%
Category:	Login	**Estimated time:**	0.00 hour
Target version:	2020-07-01 Sprint		

Description

The current Google login implementation uses OpenID Connect, but it's hardwired to use the Google endpoint, and it uses the Google People API to look up alternate email addresses.

This feature adds config keys to specify an OpenID Connect endpoint as the login provider.

```
Clusters:
  zzzzz:
    Login:
      OpenIDConnect:
        Enable: true
        Issuer: https://accounts.example.com
        ClientID: aaaaaaaaaaa
        ClientSecret: zzzzzzzzzzz
```

There's no user-facing chooser page: only one (Google or generic OIDC endpoint) can be configured at a time.

Implementation:

- rename googleLoginController to oidcLoginController
- use client ID/secret from whichever set of config keys (OpenIDConnect or Google) is in play
- if using OIDC keys, don't attempt the Google People API lookup

Subtasks:

Task # 16461: Review 16171-oidc-config	**Resolved**

Related issues:

Related to Arvados Epics - Story #15322: Replace and delete sso-provider	**Resolved**	**03/11/2020**	**08/26/2020**

Associated revisions

Revision eadb9455 - 06/08/2020 02:17 PM - Tom Clegg

Merge branch '16171-oidc'

refs #16171

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

Revision 6141cb8d - 06/08/2020 02:30 PM - Tom Clegg

Merge branch '16171-oidc'

refs #16171

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

Revision 664b5469 - 06/17/2020 04:35 PM - Tom Clegg

Merge branch '16171-oidc-config'

closes #16171

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

History

#1 - 03/04/2020 04:59 PM - Tom Clegg

- Related to Story #15322: Replace and delete sso-provider added

#2 - 03/06/2020 07:42 PM - Peter Amstutz

Also need to support standard claims:

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

#3 - 05/18/2020 06:16 PM - Peter Amstutz

- Target version set to 2020-06-03 Sprint

#4 - 05/20/2020 03:50 PM - Peter Amstutz

- Assigned To set to Tom Clegg

#5 - 06/01/2020 01:58 PM - Tom Clegg

- Status changed from New to In Progress

- Description updated

#6 - 06/01/2020 03:20 PM - Tom Clegg

16171-oidc @ 15dbaf151a72d5cfc00b4ea4b4bcb64c7ed9ac14 -- https://ci.arvados.org/view/Developer/job/developer-run-tests/1881/

#7 - 06/03/2020 02:12 PM - Peter Amstutz

Tom Clegg wrote:

> 16171-oidc @ 15dbaf151a72d5cfc00b4ea4b4bcb64c7ed9ac14 -- https://ci.arvados.org/view/Developer/job/developer-run-tests/1881/

- UserAuthenticate & getAuthInfo -- holy single line argument list, Batman
- Needs a companion branch to enable in Workbench2
- We should support the "preferred_username" claim (https://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) -- I can double check that's what the customer uses to supply the unix username.

#8 - 06/03/2020 02:35 PM - Peter Amstutz

Peter Amstutz wrote:

> - We should support the "preferred_username" claim (https://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) -- I can double check that's what the customer uses to supply the unix username.

Initial customer response:

"I think your assumption is not correct. I just checked how it looks in our current AzureAD config and there this field [preferred_username] doesn't even exist.

I would suggest that this is configurable with a reasonable default value. It's also what I was told so far that the claims in the JWT token are configurable."

#9 - 06/03/2020 03:48 PM - Tom Clegg

- Target version changed from 2020-06-03 Sprint to 2020-06-17 Sprint

#10 - 06/03/2020 07:53 PM - Peter Amstutz

- Target version deleted (2020-06-17 Sprint)

Based on customer feedback I think we want the following configuration knobs:

- the field name for the primary email address (default: 'email')
- optional field name for an alternate email address (default: none)
- optional field name to get the username (default: 'preferred_username', else otherwise Arvados generates its own from the email address and/or firstname/lastname).
- if 'email_verified' is not in the response, must treat it as 'true' (or provide configuration knob to control this behavior)

#11 - 06/03/2020 07:53 PM - Peter Amstutz

- Target version set to 2020-06-17 Sprint

#12 - 06/04/2020 03:32 PM - Tom Clegg

- UserAuthenticate & getAuthInfo -- holy single line argument list, Batman

Moved more of the oauth2 config stuff into the setup func, and removed a couple of args from getAuthInfo.

In the process I noticed the config was using the Login.Google config section instead of ctrl.ClientID, so I fixed that and added a test.

That test revealed that the OIDC library is sensitive about trailing "/" in the issuer URL, so I added a bit to strip the "/" that our config loader adds. There are possible cases that can only be fixed in the oidc library by doing a more rigorous URL comparison ("https://example:443" is equivalent to "https://example/", etc) but in the meantime this handles the test fake and Google (which strip the trailing slash) and Azure Active Directory (which has a non-empty issuer path containing a tenant ID).

I'll note this in the config doc to help reduce operator annoyance about this. Not sure whether to prioritize submitting a patch upstream.

We could create a new URL type that preserves the absence of bare slash in "https://example". This should make it possible to use any endpoint no matter how it spells itself, as long as it uses the same spelling every time. It wouldn't be a full solution (it would still require the operator to use the exact same spelling as the issuer, so it would still be needlessly finicky and would still break if an issuer changes to an equivalent spelling/encoding). It seems like the proper solution is for go-oidc to do a real "equivalent URL" test instead of a string comparison. Haven't yet found a go pkg that does an rfc3986 equivalence test, though.

The Azure docs don't seem to use trailing slashes consistently.
https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-convert-app-to-be-multi-tenant:

> A single tenant application normally takes an endpoint value like: https://login.microsoftonline.com/contoso.onmicrosoft.com
> Each Azure AD tenant has a unique issuer value of the form: https://sts.windows.net/31537af4-6d77-4bb9-a681-d2394888ea26/

Hopefully the actual service is consistent with the docs in any given case.

16171-oidc @ cd3f543b2ea20a7ac5851c118d5189df080207f2 -- https://ci.arvados.org/view/Developer/job/developer-run-tests/1890/

- Needs a companion branch to enable in Workbench2

I don't think Workbench2 actually does anything with this information, but I've updated the config struct/mock for completeness.

16171-oidc arvados-workbench2|1f4bc2074e41d1e6ec0f91d4a7d0e543020d523d

#13 - 06/04/2020 06:21 PM - Tom Clegg

Tom Clegg wrote:

> We could create a new URL type that preserves the absence of bare slash in "https://example". This should make it possible to use any endpoint no matter how it spells itself, as long as it uses the same spelling every time.

...or just use the string type:

16171-oidc @ 3b4bb3d393adc3bd3ddfb4442a65087275a5c5c3 -- https://ci.arvados.org/view/Developer/job/developer-run-tests/1891/

#14 - 06/08/2020 01:50 PM - Peter Amstutz

Tom Clegg wrote:

> Tom Clegg wrote:
>
> > We could create a new URL type that preserves the absence of bare slash in "https://example". This should make it possible to use any endpoint no matter how it spells itself, as long as it uses the same spelling every time.
>
> ...or just use the string type:
>
> 16171-oidc @ 3b4bb3d393adc3bd3ddfb4442a65087275a5c5c3 -- https://ci.arvados.org/view/Developer/job/developer-run-tests/1891/

This LGTM.

#15 - 06/08/2020 01:55 PM - Peter Amstutz

- Needs a companion branch to enable in Workbench2

I don't think Workbench2 actually does anything with this information, but I've updated the config struct/mock for completeness.

16171-oidc arvados-workbench2|1f4bc2074e41d1e6ec0f91d4a7d0e543020d523d

Turns out this probably isn't necessary, the default behavior is log in via 3rd party:

```
const requirePasswordLogin = (config: Config): boolean => {
    if (config && config.clusterConfig) {
        return config.clusterConfig.Login.LDAP.Enable || config.clusterConfig.Login.PAM.Enable || false;
    }
    return false;
};
```

#16 - 06/09/2020 05:53 PM - Tom Clegg

16171-oidc-config @ 11f80ed98b70be5379abe18f1b645ab3958d078b -- https://ci.arvados.org/view/Developer/job/developer-run-tests/1906/

16171-oidc-config @ f4750d53482ddb3990426563bb424f72790b9090 -- https://ci.arvados.org/view/Developer/job/developer-run-tests/1914/

with new configs EmailClaim, EmailVerifiedClaim, UsernameClaim

#17 - 06/16/2020 07:43 PM - Lucas Di Pentima

Some minor comments:

- Typo on comment at lib/config/generated_config.go line 589
- I think is worth mentioning these additional config knobs on the documentation, maybe by just saying that there're more than the provided example and pointing to the config reference page, wdyt?

Other than that, it LGTM.

#18 - 06/17/2020 03:32 PM - Tom Clegg

- Target version changed from 2020-06-17 Sprint to 2020-07-01 Sprint

#19 - 06/17/2020 04:36 PM - Anonymous

- % Done changed from 0 to 100

- Status changed from In Progress to Resolved

Applied in changeset arvados|664b5469124c6936733ce6544393f3883b86a32f.

#20 - 10/07/2020 02:11 AM - Peter Amstutz

- Release set to 25