RBAC Design Doc #620

New issue

Closed **klizhentas** opened this issue on Nov 27, 2016 · 17 comments

klizhentas commented on Nov 27, 2016 • edited Contributor

Multiple users expressed interest in adding more fine-grained authorization to the Teleport.
This document is a proposal for RBAC - role based access control.

Current State

The Teleport user identity is stored in a certificate as KeyID field and `AllowedLogins`. And the SSH login (AKA "OS login") this user is allowed to assume is translated to `Valid Principals` in the SSH certificate for SSH compatibility purposes.

It is worth noting that we can't use custom user cert extensions to store metadata to keep its certificates compatible with OpenSSH due to this bug

RBAC Proposal

Role

A Role is a collection of permissions.

Role Object

Here's a structure of a role object:

```
# Teleport Objec type/kind: "role" in this case.
kind: role
version: v1

#
# Metadata applies to a specific role instance
#
metadata:
  # Mandatory field. First character cannot be NOT alphanumeric. The reason for this
  # is: we can create special "internal" and hidden roles with names starting with,
  # say, '#' or '@' or '_'
  name: admin
  description: "this Role is an example"

#
# Spec is a "frozen" part of a role definition (API contract) and it is not instance-specif
#
spec:
  # Maximum allowed SSH session duration (would be nice to also enforce it on web logins)
  max_session_ttl: 3h

  # List of local OS logins allowed for this role. Similar to (obsolete) --local-logins fla
  logins: [vagrant, root]

  # Nodes this user can login into. The "label spec" is the same format as "tsh ssh" (Telep
  # parses and validates "label spec"). By default "*" which means "any node"
  node_labels: {*: login, os: *}

  # Namespaces this user is allowed to see.
  namespaces: [default, staging]

  # Permissions is a list of non-SSH permissions that apply to Teleport-specific objects
  permissions:
    sessions: [read, write]
```

Cluster namespaces

Assignees

kontsevoy

Labels

documentation
enhancement
P1

Projects

None yet

Milestone

2.0

Notifications

Subscribe

You're not receiving notifications from this thread.

6 participants

In addition to clusters we are going to introduce namespaces to limit access and visibility for various teams within the same cluster.

Teleport nodes will belong to namespaces, by default teleport node will be provisioned in `default` namespace, however we can provision nodes in different namespace within the same cluster.

Sessions and all activity executed on the node within the namespace will be automatically assigned to the same namespace.

Teleport users and roles

Teleport user is assigned to multiple roles.

User object will get new property `roles` with a list of roles assigned to it.

```
roles:
    - @bob
    - db-admin
    - readonly
```

Every user will have a default role created and associated with this user only, e.g. `alice` will get role `@alice`.

If user is assigned multiple roles, resulting permissions set is a union of all roles assigned to the user.

Migration for existing users

Existing users will be migrated to get to auto-generated role:

```
# unique role ID
id: @bob
kind: role
version: v1

spec:
   permissions:
       - {ssh: {logins: {bob: true}}}
```

This means that after the upgrade to new version of teleport with RBAC, there won't be any situation where users will be not assigned to any of the roles, simplifying management.

External users and roles

Let's assume we have an external org "Ops" that wants to get access to database nodes of the org "Bank" for db administration, but nowhere else.

1. Add trusted external cert authority `ops`
2. Add role for this new organization if necessary, or reuse existing role

```
id: ops
permissions:
   - {ssh: {logins: {bob: true}}}
```

3. Trusted certificate authority for organization "Ops" will have a `roles` property set to `ops`:

```
kind: CertificateAuthority
spec:
   roles: {ops: true}
```

Note that users for external org do not have to sync user entries on the cluster "Bank", however locally, in cluster "Org" they will have to be assigned to the role "ops" to get a certificate signed by certificate authority "ops" and get access.

Changes to the CLI

Adding users

Admins will be able to specify the role when creating new local users.

```
tctl user add bob --role=admin
```

Also note, that `--logins` flag will be still supported, to maintain backwards compatibility, we can allow using this flag, that will simply update a role associated with this user `@bob`

Adding roles dynamically

I suggest we allow users to add/update roles via CLI:

```
tctl upsert -f role.yaml
```

we will reuse `upsert` command extending support for other types of objects, e.g. users and certificate authorities.

Changes to the configuration

Some roles may be supplied statically for static teleport backend use-case:

```
auth:
   roles:
      admin:
          permissions:
              ... spec goes here
```

Mapping OIDC/OAuth2.0 claims to Roles

We are going to update `OIDC` connectors with special dynamic mappings to map claims to roles defined in the system.

Consider the following OIDC connector:

```
- id: google
  redirect_url: https://localhost:3080/v1/webapi/oidc/callback
  client_id: id-from-google.apps.googleusercontent.com
  client_secret: secret-key-from-google
  issuer_url: https://accounts.google.com
  # will add additional claims to request in auth request
  scope:
     - group
  # map group claim values to teleport roles
  # returned claim value could be list object ["a", "b"] or a simple string
  claims_to_roles:
     - {claim: 'group', 'value': "admin", role: "teleport-admin"}
```

klizhentas added **documentation** **enhancement** labels on Nov 27, 2016

kontsevoy was assigned by **klizhentas** on Nov 27, 2016

This was referenced on Nov 27, 2016

Ability to specify which users can login to which nodes #616 Closed

Automatically create a teleport user when using OIDC #624 Closed

tehsis commented on Dec 1, 2016 • edited Contributor +

Are you already working on this? is there some wip I can take a look at? is there an estimated time for this to land in master?

klizhentas referenced this issue on Dec 1, 2016

Create user with single-factor authentication? #628 Closed

klizhentas commented on Dec 8, 2016 Contributor +

Feedback from Ev:

- Create implicit super admin role that is implicitly present on all nodes
- Create a special CA that associates this user to this admin role.

This will allow us to retain admin access even in case if auth server is down.

kontsevoy commented on Dec 9, 2016 Contributor

@tehsis see ablove, this is slowly getting into shape

brumfb commented on Dec 9, 2016 • edited

Some questions on this:

1. My assumption is that roles will persist in whatever storage mechanism is selected, can you confirm?
2. If roles are added directly to the storage mechanism, will it require a reboot of the service?
3. Can static roles be combined with dynamic roles?
4. Can roles include other roles?
5. What is the role merge policy for potentially conflicting attributes? (e.g. max_session_ttl 2hr vs 3hr?)

klizhentas commented on Dec 9, 2016 • edited Contributor

See my answers below:

> My assumption is that roles will persist in whatever storage mechanism is selected, can you confirm?

Yes

> If roles are added directly to the storage mechanism, will it require a reboot of the service?

No, they will get fetched by nodes automatically from the auth service

> Can static roles be combined with dynamic roles?

If by static you mean the ones present in configs then yes

> Can roles include other roles?

No, but you will be able to have a user that is assigned multiple roles at once, in this case the resulting role will be computed using union method.

> What is the role merge policy for potentially conflicting attributes? (e.g. max_session_ttl 2hr vs 3hr?)

In this case more permissive attribute will take over - you can always restrict it back by removing the role from this user.

klizhentas referenced this issue on Dec 14, 2016

"With google" login hanging and seeing "go-oidc: provider config sync failed" in our logs #639 Closed

klizhentas added this to the **1.5** milestone on Dec 19, 2016

This was referenced on Dec 21, 2016

Configurable flag to allow read/only or private sessions #501 Closed

Is ACL/RBAC available? #478 Closed

Sasha/rbac #652 Merged

klizhentas commented on Dec 24, 2016 Contributor

@tehsis hey Pablo, can you help me out trying out Auth0 with OIDC?

I'm having trouble setting up OIDC with Auth0. I'm getting error when trying verify JWT signature: {"message":"OAuth2 error code=unsupported_response_type, message=unable to verify auth code with issuer"}

The signature in JWT token does not match fetched public key. Here's my OIDC config:

```
oidc_connectors:
- id: auth0
redirect_url: https://localhost:3080/v1/webapi/oidc/callback
client_id: kzcs2YAW6afGXn2r5KAFtt85BWx0kZJj
client_secret: <secret>
issuer_url: https://gravitational.auth0.com/
```

Is there anything I'm missing ? Can you post OIDC config that worked for you?>

klizhentas added a commit that referenced this issue on Dec 24, 2016

`work in progress, last part of #620` `5be8bd3`

klizhentas added a commit that referenced this issue on Dec 25, 2016

`map OIDC scopes to roles, implements #620` `8ab3add`

klizhentas referenced this issue on Dec 25, 2016

map OIDC scopes to roles, implements #620 #662 Merged

kontsevoy commented on Dec 26, 2016 Contributor

@klizhentas shall I close this?

kontsevoy added the P1 label on Dec 26, 2016

klizhentas commented on Dec 27, 2016 Contributor

all implemented, closing.

klizhentas closed this on Dec 27, 2016

therc commented on Dec 29, 2016

Nitpick: does the OIDC connector mapping in the example actually work? When I read it, my heart almost skipped a beat. As far as I know, Google's tokens don't support any kind of group membership claims. :-(

klizhentas commented on Dec 29, 2016 • edited Contributor

@therc Google's OIDC does not support any custom membership claims, however we are going to demo the workaround that uses identity federation + some simple plugins to inject custom claims in OIDC for google or github

vancluever commented on Jan 21

Question if anyone is watching this thread still? Was `roles` in configuration actually implemented? I'm looking at the `config.Auth` struct and can't see any reference to the `roles` YAML subkey.

klizhentas commented on Jan 21 Contributor

@vancluever this did not end up in static config, but you can try out alpha `v2.0.0-alpha.0` that has `tctl upsert -f role-spec.yaml command

klizhentas commented on Jan 21 Contributor

although we don't recommend to use alpha in production.

vancluever commented on Jan 21

@klizhentas thanks! I'm mainly looking for a way to ensure that OIDC users get auto-added in a way that I can control access purely thru configuration, so command line is not necessarily what I'm looking for here. Although bootstrapping the roles via config is something that I may play with, as I could see there being a path to doing that on startup of the service.

vancluever commented on Jan 22

@klizhentas I think I got a pattern here to pre-loading roles via `tctl upsert` on launch of teleport which will satisfy our needs.

Question: how volatile do you think (not expecting any guarantees of course) the rolespec, or `claims_to_roles` will be leading up to the v2.0 release? Will there a be any significant changes or massive improvements to things that would affect an entry point (more than likely would be containerizing this) that would do the following:

- Teleport launches in the background (in case a bootstrap is needed)
- Roles are loaded from a pre-determined directory if it exists (ie: `/etc/teleport/roles.d/`)
- Teleport process is foregrounded to allow for signal handling and CTRL-C

kontsevoy commented on Jan 22 • edited Contributor

@vancluever I will let **@klizhentas** to answer your questions, but it looks to me you're missing one aspect of Teleport design: `tctl upsert` adds stuff to a database, AKA secrets backend [1], for future use, i.e. you need to "upsert" your roles just once, no need to re-creating them on teleport.

[1] Supported backends for storing secrets (including roles) are: BoltDB, directory, etcd and DynamoDB.

vancluever commented on Jan 22

@kontsevoy yeah I get the fact that it's a database operation, but what I'm looking for (as what was originally implied from the design doc here) is the ability to completely control the configuration of Teleport, including its auth roles, via configuration. Having the ability to automate the upsert of roles from a static configuration means that I can maintain roles in configuration management, along with the Teleport configuration at large save pertinent sensitive information such as registration tokens and OAuth access info.

Given that the use case I'm targeting is OIDC, I think that I could (and would like to) get away with as little manual management as possible. I still plan on using one of the secret backends, and making sure clustering is handled properly, but seeing as an upsert operation is, well, an upsert operation, I don't see too much harm in loading configuration every startup cycle (which hopefully won't be *too* often).

This was referenced on Feb 2

Feature Request: Better Access Controls #743 Closed

feature question: does one instance of teleport support multiple clusters #781 Closed

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported Comment