[SIP] Proposal for Restricting access to Dashboards

Motivation

Dashboards might have PII or sensitive data. Currently anyone who has access to Superset can look at any Dashboard by using dashboard_id directly. They might not get access to the charts or underlying data, but can still look at the Dashboard, its structure, underlying errors etc.

Currently, anyone can access any Dashboard

- Users can bookmark dashboards and access them later
- They can share Dashboard ID and/or URL directly to anyone
- Users can directly type in the dashboard id in the form superset/dashboard/<dashboard ID>
- Even if they do not know of any particular dashboard ID, they can iterate through the dashboard to find the one they are looking for.

Once user gets the ID and hence the dashboard:

- They will be able to look at the Dashboard, layout, any headers or text on the dashboard.
- **However, permission to Underlying dataset is needed to render the dashboard data and charts.**
- The concern is that anyone can get to know the layout and underlying table/bucket/hive/chart details, even if they do **not** actually see the data/chart.

Proposed Change

Enforce that only Owners, Co-Owners or Viewers (and Admin) can view any Dashboard

1. **Admin** users will be allowed to view **any** dashboard **as per existing functionality**
    a. Admin users permission will be provided only based on request
    b. Viewing Data/Content/Chart on the dashboard is as per permission for Admin (i.e. they can see the layout only unless they have permission for underlying data which is in the bucket)
    c. Irrespective of Published/not published
2. Anonymous users ("nobody"superset user - we have turned off this feature in our deployment but just in case) will **not** be able to see **any** dashboard
3. Logged in user should be in **owner** or **viewer** list to see dashboard, irrespective of how they land on the dashboard page.
    a. If they click on the dashboard from the Dashboards panel (possibly a stale page)
    b. From url link in a email or message or favorite or bookmark etc.
    c. Directly typing in the dashboard id (a number) – e.g. /superset/dashboard/<dashboard ID>
4. If a user has neither Owner nor Viewer privilege as defined in the Dashboard, they will get an error message
    a. TBD. Should they get a JSON error message or some UI
5. **3 & 4 is a breaking change, meaning, we are changing the default behavior. If some user or customer relied on this, they will now start seeing error message until the owner of the Dashboard adds them to Viewers (or Owners). We do not want to have this under a feature switch, due to Security concern.**
6. **3 & 4 will provide additional protection to Dashboard PII data**

New or Changed Public Interfaces

```
    @has_access
    @expose("/dashboard/<dashboard_id>/")
    def dashboard(self, dashboard_id):
        """Server side rendering for a dashboard"""

        def check_owner_or_viewer(obj):
        #See if current user has either owner or viewer permission

            if not obj:
                return False

            if g.user.is_anonymous:
                return False

            roles = [r.name for r in get_user_roles()]
            if "Admin" in roles:
                return True

            owners = []
            owners += obj.owners

            owners += obj.viewers

            owner_names = [o.username for o in owners if o]


            if g.user and hasattr(g.user, "username") and
g.user.username in owner_names:
                return True



            return False


        session = db.session()
        qry = session.query(models.Dashboard)
        if dashboard_id.isdigit():
            qry = qry.filter_by(id=int(dashboard_id))
        else:
            qry = qry.filter_by(slug=dashboard_id)

        dash = qry.one_or_none()
        if not dash:
            abort(404)
```

```
        if check_owner_or_viewer( dash ) == False:
            bootstrap_data = {
                "user_id": g.user.get_id(),

                "user_name": g.user.username,
                "user.first_name": g.user.first_name,
                "user.last_name": g.user.last_name,

                "dashboard_id": dash.id,
                "dashboard_title": dash.dashboard_title,
                "error": "Need either Owner or Viewer privilege to
view this dashboard",
            }


            flash(__("You have no permission to view this
dashboard"), "danger")

            return json_success(json.dumps(bootstrap_data))




        datasources = set()
        for slc in dash.slices:
            datasource = slc.datasource
            if datasource:
                datasources.add(datasource)
```

New dependencies

No new dependency.

Migration Plan and Compatibility

No migration step involved

Rejected Alternatives

None