**VAULT POLICIES**

**Task :**

User Murali is restricted to s3 role. Murali is a part of orange group.

User Pavan is restricted to ec2 role. Pavan is a part of apple group.

**Pre-requisites :**

You should have your aws secret engine enabled and then have 2 roles – s3user and ec2-role.

**Step – 1:**

Enable user pass auth method and create the users. When you login entities will be created for the users for the first login. Each client that logs into the vault will have an entity created.

**Step-2 :**

Create 2 policies

**Apple policy with ec2 restricted access:**

# Allow tokens to look up their own properties

path "auth/token/lookup-self" {

capabilities = ["read"]

}

# Allow tokens to renew themselves

path "auth/token/renew-self" {

capabilities = ["update"]

}

# Allow tokens to revoke themselves

path "auth/token/revoke-self" {

capabilities = ["update"]

}

# Allow a token to look up its own capabilities on a path

path "sys/capabilities-self" {

capabilities = ["update"]

}

# Allow a token to look up its own entity by id or name

path "identity/entity/\*" {

capabilities = ["list"]

}

path "identity/entity/id/3d5d2efa-c02d-1e47-de1d-f71c5bf9ac20" {

capabilities = ["read","list","update","create","delete"]

}

# Allow a token to look up its own entity by id or name

path "sys/auth" {

capabilities = ["list","read"]

}

# Allow a token to renew a lease via lease\_id in the request body; old path for

# old clients, new path for newer

path "sys/renew" {

capabilities = ["update"]

}

path "sys/leases/renew" {

capabilities = ["update"]

}

# Allow looking up lease properties. This requires knowing the lease ID ahead

# of time and does not divulge any sensitive information.

path "sys/leases/lookup" {

capabilities = ["update"]

}

path "aws/\*" {

capabilities = ["list"]

}

path "aws/roles/ec2-role"{

capabilities = ["list","delete","create","update","read"]

}

path "aws/creds/ec2-role" {

capabilities = ["read","list","update","create","delete"]

}

**Orange policy with s3 restricted access :**

path "aws/\*" {  
  capabilities = ["list"]  
}

path "aws/roles/s3user" {  
  capabilities = ["list","read","update","delete","create"]  
}

path "aws/creds/s3user" {  
  capabilities = ["list","read","update","create"]  
}

# Allow a token to look up its own capabilities on a path  
path "sys/capabilities-self" {  
    capabilities = ["update"]  
}

path "sys/policies/acl"  
{  
capabilities = ["list"]  
}

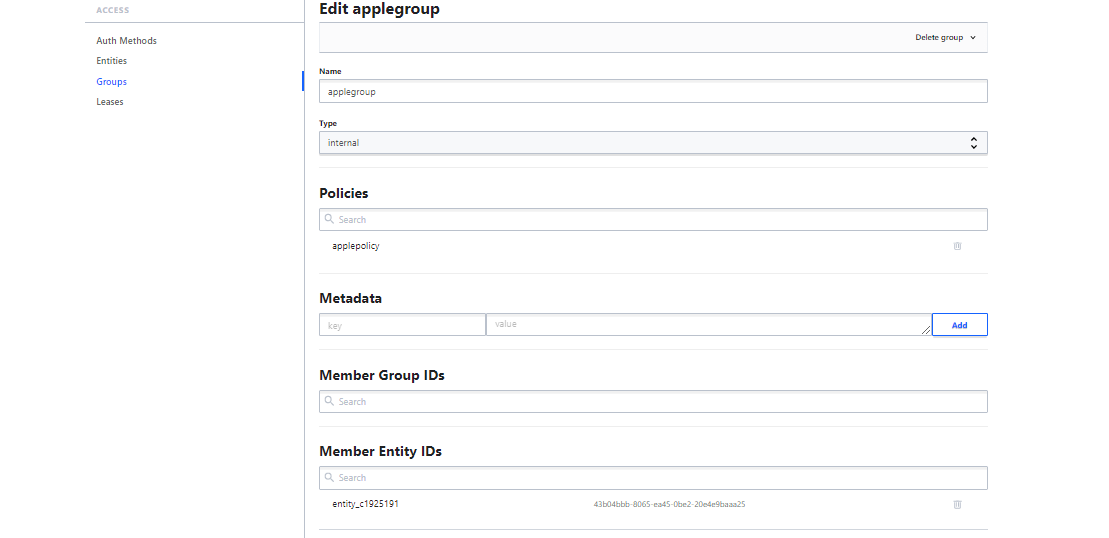
**Step-3 :**

Create 2 groups 🡺 Apple group and orange group

In apple group add pavan and attach apple policy

In orange group add murali and attach orange policy to it .

The following screenshot is for apple group similarly you can configure orange group.



**Default policy :**

# Allow tokens to look up their own properties

path "auth/token/lookup-self" {

capabilities = ["read"]

}

# Allow tokens to renew themselves

path "auth/token/renew-self" {

capabilities = ["update"]

}

# Allow tokens to revoke themselves

path "auth/token/revoke-self" {

capabilities = ["update"]

}

The above 3 paths will allow the user to revoke/renew the token

# Allow a token to look up its own capabilities on a path

path "sys/capabilities-self" {

capabilities = ["update"]

}

The above path will help you see what capabilities you have on a specific path or a role.

# Allow a token to look up its own entity by id or name

path "identity/entity/id/{{identity.entity.id}}" {

capabilities = ["read"]

}

path "identity/entity/name/{{identity.entity.name}}" {

capabilities = ["read"]

}

The above 2 paths will help you restrict access to entities that you can see.

# Allow a token to look up its resultant ACL from all policies. This is useful

# for UIs.

path "sys/policies/acl"  
{  
capabilities = ["list"]  
}

The user will be able to list the policies present.