Dynamic Policy for Access Control

Adam Young

Senior Software Engineer, Identity Management

Red Hat

<ayoung@redhat.com>

Westford, MA

18 MAY 2015

Agenda

1 Authorization in OpenStack

2 Defaults

3 Custom Policy

4 In Development

5 Vision

6 Conclusion

Who am I to talk about Keystone?

- Contributor Since Diablo
- Core Since June 2012
- LDAP
- PKI Tokens
- Kerberos
- Trusts

Keystone is My Day Job

Section 1

Authorization in OpenStack

Cloud Scale

Question: How do you manage several datacenters
with thousands of physical virtual machines
each running dozens of virtual machines?

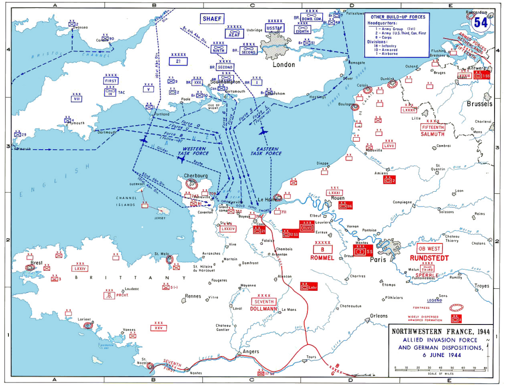
Answer: Delegate as much as Possible

Definitions of Delegate[verb]

- The assignment of responsibility or authority to another person to carry out specific activities.
- The practice of effectively getting others to perform work which one chooses not to do oneself.
- Entrust a task or responsibility to another person

Cloud Scale

If you are working at scale



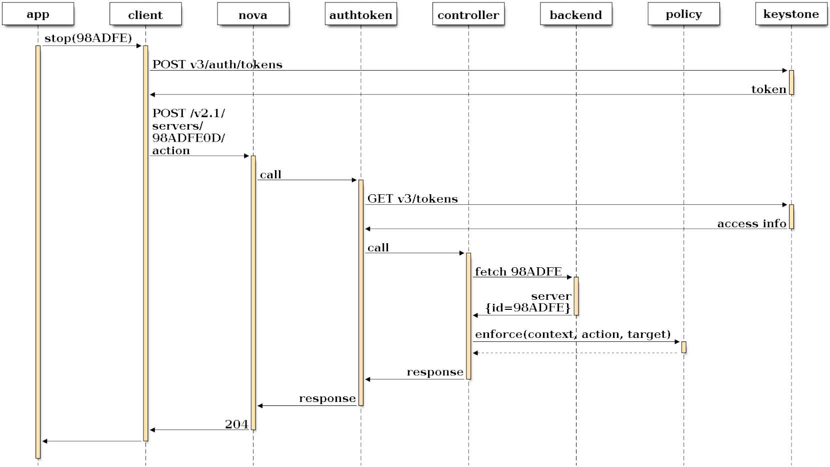
Cloud Scale

Delegation is essential to getting the Job Done

OpenStack Access Control

- (Scoped) Role Based Access Control (RBAC)
- Implemented via Attribute Based Access Control
- User or Group Assigned Role on Project
- Access is Checked per API on Remote Service

API Authorization Sequence



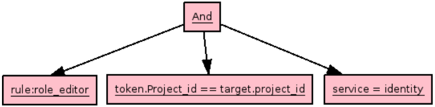
keystonemiddleware.auth_token

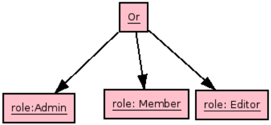
- In the middleware stack for endpoints
- Validates token
- Rejects an invalid token.
- Expands access information
- Adds headers to the context
- Config option to let request without token pass
- Does not enforce policy

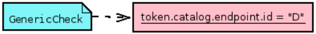
Oslo Policy

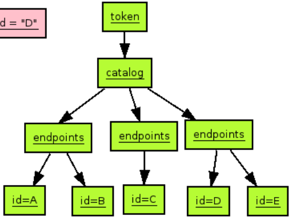
- Rules Engine
- Access? Yes or No.
- Uses token AccessInfo + Requested Resources
- Each Endpoint decides when to call
- Was in incubated
- Promoted to Oslo.policy library
- Only Keystone uses library in Kilo

Policy Rules

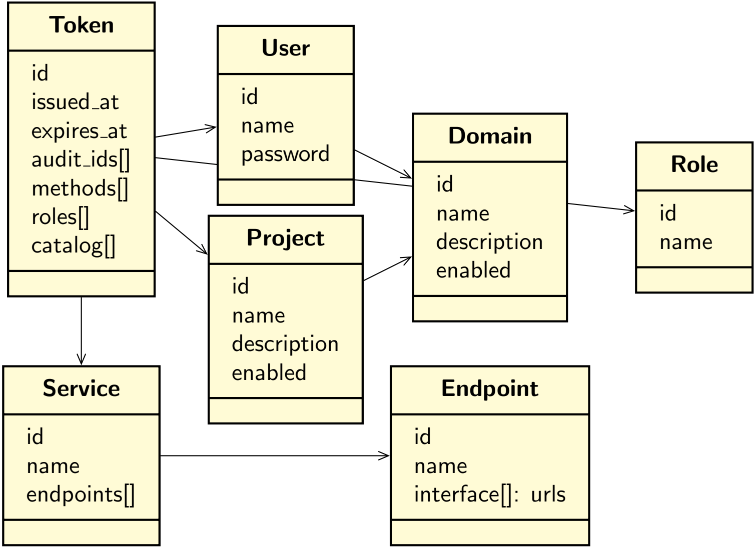








What comes from a Token



Section 2

Defaults

Bug 968696

"admin"-ness not properly scoped

- reported by Gabriel Hurley on 2012-03-29
- Roles were Global
- Role:admin
- Backwards Compatibility

Nova's Default Policy File:

- Common Rules
  - "context_is_admin": "role:admin",
  - "admin_or_owner": "is_admin:True or project_id:
  - "default": "rule:admin_or_owner",
- Examples
  - "compute:create": "",
  - "admin_api": "is_admin:True",
  - "compute:v3:servers:start": "rule:admin_or_owner",

Neutron’s Default Policy File

- Starts with Nova’s common rules
- "admin_or_network_owner": "rule:context_is_admin or tenant_id:
- "admin_only": "rule:context_is_admin",
- "regular_user": "",
- "shared": "field:networks:shared=True",
- "shared_firewalls": "field:firewalls:shared=True",
- "external": "field:networks:router:external=True",

Glance’s Default Policy File

- Header is simple
  - ”context_is_admin”: ”role:admin”,
  - ”default”: ””,
- examples
  - ”add_image”: ””,
  - ”publicize_image”: ”role:admin”,

Keystone’s default Policy File

- ”admin_required”: ”role:admin or is_admin:1”,
- ”service_role”: ”role:service”,
- ”service_or_admin”: ”rule:admin_required or rule:service_role”,
- ”owner” : ”user_id:
- ”admin_or_owner”: ”rule:admin_required or rule:owner”,
- ”token_subject”: ”user_id:
- ”admin_or_token_subject”: ”rule:admin_required or rule:token_subject”,
- ”default”: ”rule:admin_required”,

Section 3

Custom Policy

Policy.V3Cloudsample.json

- First iteration of a better approach
- Set an administrative Domain for Identity Operations
- Horizon does not support Domain Scoped tokens
- Domain Id has to match:
  - %(target.project.domain_id)
  - %(project.domain_id)
  - %(target.user.domain_id)
  - %(target.group.domain_id)
  - %(group.domain_id)
  - %(scope.domain.id)

The Seventy Maxims of Maximally Effective Mercenaries

"Maxim 30: A little trust goes a long way.
The less you trust, the further you go."
– Howard Tayler, schlockmercenary.com"

Image courtesy of Hypernode Media, www.schlockmercenary.com, used with permission.

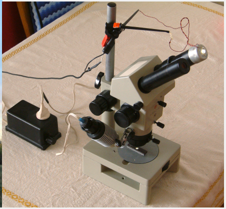
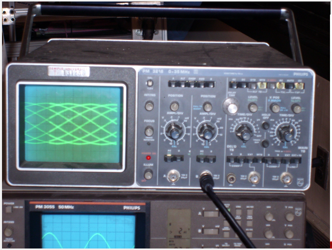
What we can do today (Juno and later)

- Merge Policy Header
- Unify Nova, Glance and Cinder
- Move toward a single File
- Break member up into smaller roles
- remove is_admin from lines
- Fetch policy via Endpoint Policy Extension

Role Composition

- Redundant to always write (role:admin or role:member)
- The Siphonaptera
- Admin role implies member privileges
- “role_member” :”(role:admin or role:member)”
- Admin -> Member -> Writer -> Reader
- Assign the lowest level and build to larger ones
- Split along resource lines
  - Network writer versus storage writer

Scope All the Things

- 
- 

via ONCEUPONAWIN.COM

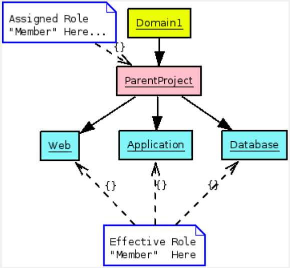
I stole this gag from the Oatmeal. I don't know who he stole it from.

Scoping in Policy

- Each API level policy rule has Scope and Role
- "compute:create": "project_id:%(project_id)s and (role:admin or role:member)"
- Explicitly Match Scope on the API
  - Where do you find project_id
  - Remember policy.v3cloudsample.json?
- How do you get a token for the right scope?

Hierarchical Multitenancy

- Nested Projects
- Role Assignment Inheritance
  - assigment in parent project **may be** inherited when getting a token in a subordinate project
  - Role is either on parent OR all children
- New in Kilo



Who can do what?

- Given a token, what APIs can I perform?
- https://review.openstack.org/#/c/170978/
- Assumes target project_id matches scope of token
- Horizon does this now, but with copies of the policy files.

Section 4

In Development

Mission

Task

Secure Delegation

Design Goals

- Simplify
- Customize
- Manage Risk

What Problems are we solving

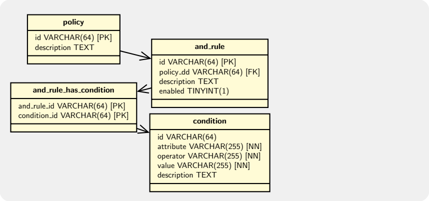
- Minimize Attack Surface.
- Determine what roles they need to perform some action
- Delegate a subset of their capabilites to a remote service.
- Determine capablities from roles
-

Policy Distribution

- Enforce Policy from keystonemiddelware
- Fetch the policy.json file from Keystone.
- default if not explicitly assigned for an endpoint

Policy Generation

- Single Default Policy file
- Database schema to hold the rules from the policy file
- Composed (hierarchical) role definitions
- Break member up into smaller roles.



policy
id VARCHAR(64) [PK]
description TEXT

and_rule
id VARCHAR(64) [PK]
policy_dd VARCHAR(64) [FK]
description TEXT
enabled TINYINT(1)

and_rule_has_condition
and_rule_id VARCHAR(64) [PK]
condition_id VARCHAR(64) [PK]

condition
id VARCHAR(64)
attribute VARCHAR(255) [NN]
operator VARCHAR(255) [NN]
value VARCHAR(255) [NN]
description TEXT

Guidance

- Better Sample Policies
- Finer grained Roles
- Better Matches of Roles to APIs
- Enforce on Keystone V3 Tokens Only

Section 5

Vision

Unified Delegation

- Must have a role assigned yourself in order to assign it.
- Track Who granted the role assignment
- Deal with the boss being fired
- Use the same mechanism for
  - Role Assignments
  - OS-TRUSTS
  - OS-OAUTH1
- Assign only a subset of a role
- Precanned delegation agreements
- Cinder, Nova fetching tokens for actions on other endpoints
- Past Liberty

Section 6

Conclusion

Image Attributions

- I didn't use all these.
- http://en.wikipedia.org/wiki/Insect
- http://mediad.publicbroadcasting.net/p/nhpr/files/styles/medium/p
- http://scientistsbookshelf.org/assets/WindPower.jpg
- http://farm5.static.flickr.com/4035/4524737863_662b41039d_o.jpg
- http://2012grade10.wikispaces.com/file/view/kal.jpg/335877836/36
- http://media-2.web.britannica.com/eb-media/69/79869-050-EA54190E.jpg
- https://socialsciences.uchicago.edu/sites/default/files/styles/hero/pu

Questions?

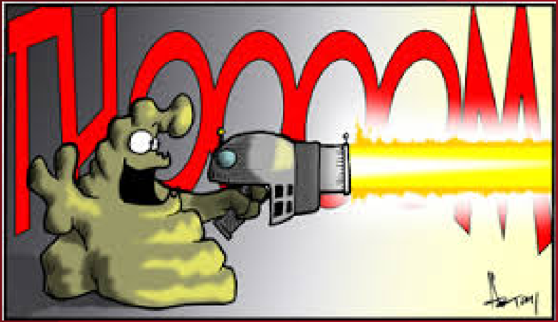


Image courtesy of Hypernode Media, www.schlockmercenary.com, used with permission.