



FOOLPROOFING YOUR OPENSTACK WITH PATROLE

Pradeep Kumar KS

Nishant Kumar

N Hemanth Kumar

Smruti Soumitra Khuntia

WARM UP FOR WORKSHOP



- › Etherpad link for

https://etherpad.openstack.org/p/BerlinSummit_Workshop_Patrole

(Please pick your machine and run pre-requisites)

- › Etherpad link for Exercises

› https://etherpad.openstack.org/p/BerlinSummit_Workshop_Patrole_Exercise

AGENDA OF WORKSHOP



- › Role-Based Access Control Overview
- › Patrole Overview
- › Hands on Demo.

ROLE BASED ACCESS CONTROL



- › Role-Based Access Control (RBAC) is used by most OpenStack services to control user access to resources. Authorization is granted if a user has the necessary role to perform an action.
 - Default Policy file : `policy.json`
 - Policy in Code
 - Custom Policy

WHY PATROLE



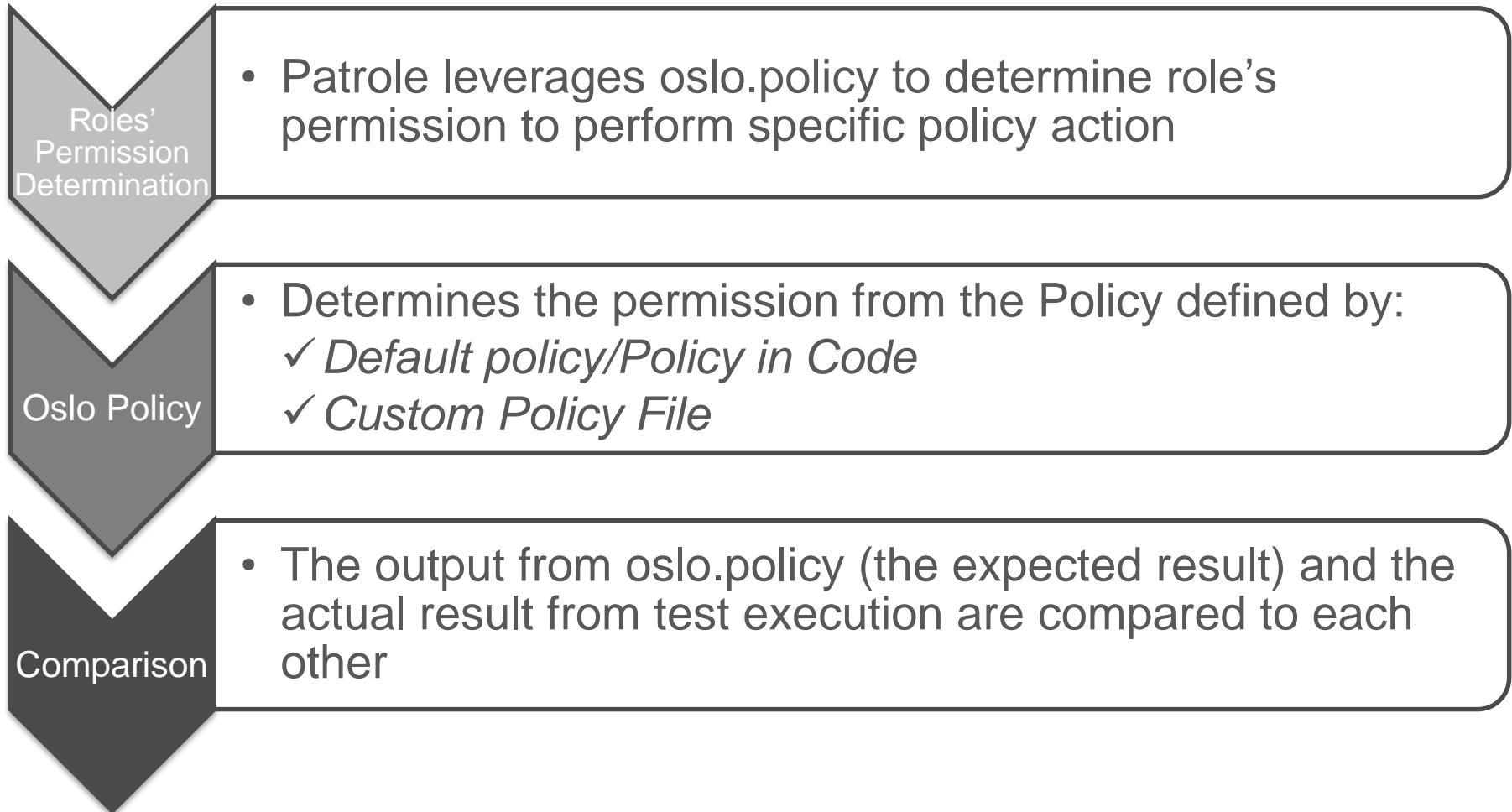
- › Patrole provides security validation process for Role based access Control enforcement in OpenStack cloud. A tool to validate that each of the resources *can* be accessed by authorized users and *cannot* be accessed by unauthorized users.
- › Patrole provides a set of integration test suits to run against live OpenStack cluster to validate correctness and integrity of cloud's RBAC implementation and policy configuration.
- › Patrole leverages Tempest based API tests using specified RBAC roles.

PATROLE: FEATURES



- › Validation of default policy definitions located in policy.json files.
- › Validation of in-code policy definitions.
- › Validation of custom policy file definitions that override default policy definitions.
- › Built-in positive and negative testing. Positive and negative testing are performed using the same tests and role-switching.
- › Validation of custom roles as well as default OpenStack roles.

PATROLE: HOW IT WORKS



PATROLE: HOW IT WORKS



Terminology

Expected Result	The expected result of a given test
Actual Result	The actual result of a given test.
Final Result	A match between both expected and actual results. A mismatch in the expected result and the actual result will result in a test failure.

Expected	Actual	Test Case	Impression
Pass	Pass	Success	
Pass	Fail	Failure	Under Permission
Fail	Pass	Failure	Over Permission
Fail	Fail	Success	Expected Exception
Fail	Fail	Failure	Unexpected Exception



DEMONSTRATION

PREREQUISITE



- › 0a. Start OpenStack services
- › 0b. Copy tempest.conf from tempest folder to default location

Etherpad link: https://etherpad.openstack.org/p/BerlinSummit_Workshop_Patrole_Exercise

EXERCISE - 1



- › 1a. List the existing plugins and tests provided by plugins
- › 1b. Update patrol section of tempest conf
- › 1c. Execute patrol test cases
- › 1d. Switch the role to reader and rerun the above test cases

Etherpad link: https://etherpad.openstack.org/p/BerlinSummit_Workshop_Patrole_Exercise

EXERCISE - 2



- › 2a. Execute the SUCCESS scenario (Expected: Pass, Actual: Pass)
- › 2b. Execute the SUCCESS scenario (Expected: Fail, Actual: Fail)

Etherpad link: https://etherpad.openstack.org/p/BerlinSummit_Workshop_Patrole_Exercise

EXERCISE - 3



- › 3a. Execute the FAILURE scenario (Expected: Fail, Actual: Pass) - OVERPERMISSION
- › 3b. Execute the FAILURE scenario (Expected: Pass, Actual: Fail) - UNDERPERMISSION

Etherpad link: https://etherpad.openstack.org/p/BerlinSummit_Workshop_Patrole_Exercise

WRITING YOUR OWN CUSTOM REQUIREMENTS FILE



```
<service_foo>:
  <logical_or_example>:
    - <allowed_role_1>
    - <allowed_role_2>
  <logical_and_example>:
    - <allowed_role_3>, <allowed_role_4>
<service_bar>:
  <logical_not_example>:
    - <!disallowed_role_5>
```

service – the service being tested

api_action – policy action being tested

volume:create

os_compute_api:servers:start

allowed_role – oslo.policy role that is allowed to perform the API.

EXERCISE - 4



- › 4a. Execute the SUCCESS scenario with Patrole Custom Requirements
- › 4b. Execute the FAILURE scenario with Patrole Custom Requirements

Etherpad link: https://etherpad.openstack.org/p/BerlinSummit_Workshop_Patrole_Exercise

IMPLEMENTING A NEW PATROLE TESTCASE



- › Things to consider while writing a new Patrole Testcase
 - Role Overriding
 - Test Setup
 - Test Execution
 - Test Cleanup

REFERENCES



- › <https://docs.openstack.org/patrole/latest/index.html>
- › <https://docs.openstack.org/tempest/latest/index.html>

HAPPY TO HELP!



- › Smruti Soumitra Khuntia : soumitra.khuntia@ericsson.com
- › Hemanth Kumar Nakkina : n.hemanth.kumar@ericsson.com
- › Pradeep Kumar KS : pradeepkumar.ks@ericsson.com
- › Nishant Kumar : nishant.e.kumar@ericsson.com



SPECIAL THANKS TO OVH FOR
ENABLING THIS WORKSHOP
WITH CLOUD INSTANCES



QUESTIONS AND ANSWERS



ERICSSON