OPNFV MOON developer guide

Release draft (51a65a7)

OPNFV

August 12, 2016

CONTENTS

CHAPTER ONE

INTRODUCTION

This guide presents the use of the MoonClient script. The MoonClient script allows the administrator/user to drive the Moon platform and some parts of the Keystone server itself.

CHAPTER

TWO

PRE-REQUISITE

Before using the MoonClient script, you must export some variables in your shell. Those variables are the same as the variables used to execute the OpenStack client or the Nova/Swift/Neutron/... clients. You can export directly in the shell like this:

```
export OS_USERNAME=admin
export OS_PASSWORD=password
export OS_REGION_NAME=What_ever_you_want
export OS_TENANT_NAME=admin
export OS_AUTH_URL=http://localhost:5000/v2.0
```

Or you can source a shell script:

```
cat << EOF | tee ~/set_openstack_auth.sh
#!/usr/bin/env bash
export OS_USERNAME=admin
export OS_PASSWORD=password
export OS_REGION_NAME=What_ever_you_want
export OS_TENANT_NAME=admin
export OS_AUTH_URL=http://localhost:5000/v2.0
EOF
source ~/set_openstack_auth.sh
```

It is important to notice that those variables are **exactly** the same you use for the OpenStack clients.

CHAPTER

THREE

USAGE

The main usage of the MoonClient is:

```
$ moon --help
usage: moon [--version] [-v | -q] [--log-file LOG_FILE] [-h] [--debug]
            [--username <username-str>] [--tenant <tenantname-str>]
            [--password <password-str>] [--authurl <authurl-str>]
            command
...
```

MoonClient ca be used interactively:

```
$ moon
```

or by using a specific command, like:

```
$ moon tenant list
```

3.1 Output formats

The default output is a tabular:

```
$ moon tenant list
+----------------------------------+-------+---------------+--------------------------+--------------
| id                               | name  | description   | intra_authz_extension_id | intra_admin_e
+----------------------------------+-------+---------------+--------------------------+--------------
| 9fb190ddb19a4a138837e2740daec3ae | admin | Admin Project |                          | None
+----------------------------------+-------+---------------+--------------------------+--------------
```

But you can have other output format, use the *-f bad_value* to see them all:

```
$ moon tenant list -f bad_value
usage: moon tenant list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
                        [--max-width <integer>] [--noindent]
                        [--quote {all,minimal,none,nonnumeric}]
moon tenant list: error: argument -f/--format: invalid choice: 'bad_value' (choose from 'csv', 'json
```

For example, the same command with a JSON output format:

```
$ moon tenant list -f json
[
  {
    "intra_authz_extension_id": "",
    "description": "Admin Project",
    "intra_admin_extension_id": null,
    "id": "9fb190ddb19a4a138837e2740daec3ae",
    "name": "admin"
  }
]
```

You can also select one or more columns with the *-c* attribute:

```
$ moon tenant list -f value -c id -c name
9fb190ddb19a4a138837e2740daec3ae admin
```

3.2 Commands

All commands can be categorized like this: * *tenant* command to get, put or delete tenants (projects in OpenStack) * *intraextension* command to get, put or delete intra_extensions in Moon * *subject object* and *action* commands to configure subject, object or action in intra_extensions in Moon * *rule* command to set rules in an intra_extension * some configuration commands (*template*, *submetarule*, *aggregation*) to configure Moon and the relation in and between intra_extensions * *log* command to show events in the Moon logging system * *test* command to run tests against the Moon platform

All commands can be prefixed with the command *help* to have information about usage of that command.

3.3 Basic example:

Here is a basic example of how you can use MoonClient:

```
$ moon tenant list

$ openstack project list
+----------------------------------+-----------------+
| ID                               | Name            |
+----------------------------------+-----------------+
| 06f2f729f5b041f290295d2d966aff00 | alt_demo        |
| 7d5fb06879ae4a0c82948d4ed7b87b7c | demo            |
| 833a954bfd314de09b09aac00f1aa647 | service         |
| 9fb190ddb19a4a138837e2740daec3ae | admin           |
| ca680df7ee10480d89414c74d46e2c65 | sdn             |
| d30cffc153b743e88a8d78052737b556 | test_moonclient |
+----------------------------------+-----------------+
$ moon tenant add admin
$ moon tenant list
+----------------------------------+-------+---------------+--------------------------+--------------
| id                               | name  | description   | intra_authz_extension_id | intra_admin_e
+----------------------------------+-------+---------------+--------------------------+--------------
| 9fb190ddb19a4a138837e2740daec3ae | admin | Admin Project | None                     | None
+----------------------------------+-------+---------------+--------------------------+--------------
$ moon intraextension list
+----------------------------------+------------------+--------------------+
| id                               | name             | model              |
+----------------------------------+------------------+--------------------+
| d508874d08424ee8a78ded9d5e008685 | policy_root      | policy_root        |
+----------------------------------+------------------+--------------------+
$ moon template list
+--------------------+-------------------+-----------------------------+
| id                 | name              | description                 |
+--------------------+-------------------+-----------------------------+
| policy_rbac_admin  | RBAC Admin Policy |                             |
| policy_root        | Root Policy       | root extension              |
| policy_authz       | Multiple_Policy   | Multiple Security Policies  |
| policy_empty_admin | Empty_Policy      | Empty Policy                |
| policy_empty_authz | MLS_Policy        | Multi Level Security Policy |
| policy_mls_authz   | MLS_Policy        | Multi Level Security Policy |
+--------------------+-------------------+-----------------------------+
$ moon intraextension add --policy_model policy_authz test
IntraExtension created: e75dad8f2f7d40de9921b0d444198973
$ moon intraextension list
+----------------------------------+------------------+--------------------+
| id                               | name             | model              |
+----------------------------------+------------------+--------------------+
| e75dad8f2f7d40de9921b0d444198973 | test             | policy_authz       |
| d508874d08424ee8a78ded9d5e008685 | policy_root      | policy_root        |
+----------------------------------+------------------+--------------------+
$ moon intraextension select e75dad8f2f7d40de9921b0d444198973
Select e75dad8f2f7d40de9921b0d444198973 IntraExtension.
$ moon intraextension show selected
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| id          | e75dad8f2f7d40de9921b0d444198973 |
| name        | test                             |
| description |                                  |
| model       | policy_authz                     |
| genre       | authz                            |
+-------------+----------------------------------+
$ moon subject list
+----------------------------------+-------+----------------------------------+
| id                               | name  | Keystone ID                      |
+----------------------------------+-------+----------------------------------+
| 04ed28e87f004ed29ddb721c43fdafb0 | demo  | 16254c7516734bca99311979f0a486bf |
| 8101e73fb82e433fbc576587e6201bfe | admin | 6b135900bf874d63abe59be074584eb9 |
+----------------------------------+-------+----------------------------------+
$ moon object list
+----------------------------------+---------+-------------+
| id                               | name    | description |
+----------------------------------+---------+-------------+
| 0fb3e294f0714a30a3b0af4c889354aa | servers | servers     |
+----------------------------------+---------+-------------+
$ openstack user list
+----------------------------------+----------+
| ID                               | Name     |
+----------------------------------+----------+
| 088758e049aa4c51bdb386fd7e954c73 | glance   |
| 16254c7516734bca99311979f0a486bf | demo     |
| 263b87f84f274260a9dbef34e7c55602 | neutron  |
| 40f2cbbe71e845b49e828b9208ba7dfc | alt_demo |
| 505a9758bd3e493f9baf44ed880aae92 | swift    |
| 62c91c632e76435a907a510ae99df378 | keystone |
| 6b135900bf874d63abe59be074584eb9 | admin    |
| b58c153e4d0647e7b61bd76d5a77916c | nova     |
| c90c58fb2aaf4fd3880a39a8d1c34263 | cinder   |
+----------------------------------+----------+
$ moon subject add test_user
Password for user test_user:
```

```
$ openstack user list
+----------------------------------+-----------+
| ID                               | Name      |
+----------------------------------+-----------+
| 088758e049aa4c51bdb386fd7e954c73 | glance    |
| 16254c7516734bca99311979f0a486bf | demo      |
| 1d2fcb31ba9b44a4bd21ae6e390ae906 | test_user |
| 263b87f84f274260a9dbef34e7c55602 | neutron   |
| 40f2cbbe71e845b49e828b9208ba7dfc | alt_demo  |
| 505a9758bd3e493f9baf44ed880aae92 | swift     |
| 62c91c632e76435a907a510ae99df378 | keystone  |
| 6b135900bf874d63abe59be074584eb9 | admin     |
| b58c153e4d0647e7b61bd76d5a77916c | nova      |
| c90c58fb2aaf4fd3880a39a8d1c34263 | cinder    |
+----------------------------------+-----------+
```

3.4 IntraExtension

An intra_extension is a module connected to a tenant/project. This connection allows to configure the authorization configuration for that tenant. The ‘intraextension‘commands has the following sub-commands:

- *add* sub-command add a new intraextension

** this sub-command needs the name of the policy template to use ** the list of all policy template can be retrieve with *moon template list* * *delete* sub-command delete an intra_extension (the deletion is definitive) * *init* sub-command must be **only** used if the root intra_extension was deleted ** the sub-command has no effect otherwise * *list* sub-command list all intra_extensions * *select* sub-commands select a specific tenant so the *–intraextension* attribute is not mandatory in other commands * *show* sub-commands print a description of the tenant given in argument ** *selected* is a special argument of the *show* sub-commands which prints the current selected tenant

There are 3 types of intra_extension:

- authz intra_extensions which are used to configure rules for standard actions (for example Nova or Swift actions)
- admin intra_extensions which are used to configure rules for authz and admin intra_extensions
- root intra_extensions which are used to configure rules for admin intra_extensions

When you start using Moon, we recommend that you only configure authz intra_extensions. Admin and root intra_extensions are already configured for your needs.

Here is an example of how to use intra_extension:

```
$ moon template list
+--------------------+-------------------+-----------------------------+
| id                 | name              | description                 |
+--------------------+-------------------+-----------------------------+
| policy_rbac_admin  | RBAC Admin Policy |                             |
| policy_root        | Root Policy       | root extension              |
| policy_authz       | Multiple_Policy   | Multiple Security Policies  |
| policy_empty_admin | Empty_Policy      | Empty Policy                |
| policy_empty_authz | MLS_Policy        | Multi Level Security Policy |
| policy_mls_authz   | MLS_Policy        | Multi Level Security Policy |
+--------------------+-------------------+-----------------------------+
$ moon intraextension add --policy_model policy_authz test
IntraExtension created: e75dad8f2f7d40de9921b0d444198973
$ moon intraextension list
+----------------------------------+------------------+--------------------+
| id                               | name              | model               |
+----------------------------------+-------------------+---------------------+
| e75dad8f2f7d40de9921b0d444198973 | test              | policy_authz        |
| d508874d08424ee8a78ded9d5e008685 | policy_root       | policy_root         |
+----------------------------------+-------------------+---------------------+
$ moon intraextension select e75dad8f2f7d40de9921b0d444198973
Select e75dad8f2f7d40de9921b0d444198973 IntraExtension.
$ moon intraextension show selected
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| id          | e75dad8f2f7d40de9921b0d444198973 |
| name        | test                             |
| description |                                  |
| model       | policy_authz                     |
| genre       | authz                            |
+-------------+----------------------------------+
```

3.5 Tenant/project

The *tenant* command allows to get information and modify projects in Keystone.

The tenant command has several sub-commands:

- *add* sub-commands add new tenant/project in Moon

** if the project doesn't exist in Keystone, it is automatically created (see example below) * *delete* sub-commands delete a tenant in Moon ** warning: it only deletes in Moon, not in Keystone * *list* sub-commands show all tenants configured in Moon ** warning it doesn't list all projects in Keystone * *set* sub-commands update a tenant in Moon ** this sub-commands is especially use to connect a tenant with an intra_extension

Here is an example of use:

```
$ openstack project list
+----------------------------------+-----------------+
| ID                               | Name            |
+----------------------------------+-----------------+
| 06f2f729f5b041f290295d2d966aff00 | alt_demo        |
| 7d5fb06879ae4a0c82948d4ed7b87b7c | demo            |
| 833a954bfd314de09b09aac00f1aa647 | service         |
| 9fb190ddb19a4a138837e2740daec3ae | admin           |
| ca680df7ee10480d89414c74d46e2c65 | sdn             |
| d30cffc153b743e88a8d78052737b556 | test_moonclient |
+----------------------------------+-----------------+
$ moon tenant list
+----------------------------------+-------+---------------+---------------------------+--------------
| id                               | name  | description   | intra_authz_extension_id  | intra_admin_e
+----------------------------------+-------+---------------+---------------------------+--------------
| 9fb190ddb19a4a138837e2740daec3ae | admin | Admin Project | None                      | None
+----------------------------------+-------+---------------+---------------------------+--------------
$ moon tenant add test_tenant
$ moon tenant list
+----------------------------------+-------------+---------------+--------------------------+--------
| id                               | name        | description   | intra_authz_extension_id | intra_a
+----------------------------------+-------------+---------------+--------------------------+--------
| 9fb190ddb19a4a138837e2740daec3ae | admin       | Admin Project | None                     | None
| 4694c91a0afb4b7d904a3bf5e886913c | test_tenant | test_tenant   | None                     | None
```

```
+----------------------------------+-------------+---------------+--------------------------+--------
$ openstack project list
+----------------------------------+-----------------+
| ID                               | Name            |
+----------------------------------+-----------------+
| 06f2f729f5b041f290295d2d966aff00 | alt_demo        |
| 4694c91a0afb4b7d904a3bf5e886913c | test_tenant     |
| 7d5fb06879ae4a0c82948d4ed7b87b7c | demo            |
| 833a954bfd314de09b09aac00f1aa647 | service         |
| 9fb190ddb19a4a138837e2740daec3ae | admin           |
| ca680df7ee10480d89414c74d46e2c65 | sdn             |
| d30cffc153b743e88a8d78052737b556 | test_moonclient |
+----------------------------------+-----------------+
```

To connect a tenant with an intra_extension, use:

```
$ moon tenant set --authz e75dad8f2f7d40de9921b0d444198973 4694c91a0afb4b7d904a3bf5e886913c
$ moon tenant list -c id -c name -c intra_authz_extension_id
+----------------------------------+-------------+----------------------------------+
| id                               | name        | intra_authz_extension_id         |
+----------------------------------+-------------+----------------------------------+
| 9fb190ddb19a4a138837e2740daec3ae | admin       | None                             |
| 4694c91a0afb4b7d904a3bf5e886913c | test_tenant | e75dad8f2f7d40de9921b0d444198973 |
+----------------------------------+-------------+----------------------------------+
```

When a tenant **is not connected to** an intra_extension, this tenant acts as a standard Keystone project. Authorization rules are evaluated by each component independently. For example, when a user ask to stop a Virtual Machine (VM), Nova

- retrieve the Keystone token and
- check its policy.json file to see if that user can stop this VM.

When a tenant **is connected to** an intra_extension, the authorisation process is driven by Moon. Authorization rules are evaluated by the Moon platform. For example, when a user ask to stop a Virtual Machine (VM), Nova

- retrieve the Keystone token
- check its policy.json file to see if that user can stop this VM and
- ask Moon if the user is authorized to do such action.

When a tenant is connected to an intra_extension, the authorisation process is driven by the following configuration.

3.6 Subject/Object/Action

The configuration of an intra_extension is mainly divided into 3 elements.

The subjects represent the users (in the future, they can also represent other elements like VM or networks). Subjects are the source of an action on an object. The objects represent the elements which is actioned by a subject (like VM, network, Swift file or directory). The actions represent what can a subject do on an object (like start a VM, create a file in Swift, ...).

Here is an example of what you can found in a standard Moon platform:

```
$ moon subject list
+----------------------------------+-----------+----------------------------------+
| id                               | name      | Keystone ID                      |
+----------------------------------+-----------+----------------------------------+
| 04ed28e87f004ed29ddb721c43fdafb0 | demo      | 16254c7516734bca99311979f0a486bf |
| 517e648cc5d64984ab18e8c76422258a | test_user | 1d2fcb31ba9b44a4bd21ae6e390ae906 |
| 8101e73fb82e433fbc576587e6201bfe | admin     | 6b135900bf874d63abe59be074584eb9 |
+----------------------------------+-----------+----------------------------------+
$ moon object list
+----------------------------------+---------+-------------+
| id                               | name    | description |
+----------------------------------+---------+-------------+
| 0fb3e294f0714a30a3b0af4c889354aa | servers | servers     |
+----------------------------------+---------+-------------+
$ moon action list
+----------------------------------+--------------+--------------+
| id                               | name         | description  |
+----------------------------------+--------------+--------------+
| e349bdad65ac43aeb1058623f9738b2b | unpause      | unpause      |
| 41b8ce4256a84f19b4322acef05f3367 | post         | post         |
| 7eea8c5b19c04d4e9cfc5a14cdd8ce84 | create       | create       |
| 78a20944dbd04b2ea33007d46bfd5ddd | download     | download     |
| a1da1466938842c2b2aace1868153192 | upload       | upload       |
| ab9f285b9670473fbe2f1501b62a2779 | list         | list         |
| b47452174c0c40a58d7cb2ba949acfe9 | storage_list | storage_list |
| 9c448d73e344472bbe189546c2c35c5d | stop         | stop         |
| b957463ac8cf4a02ad2e64c0ae38e425 | pause        | pause        |
| 3b018cd88b964e5ca69c4ef9e8045a3d | start        | start        |
+----------------------------------+--------------+--------------+
```

Note: the *servers* object is a special hardcoded object which represents all servers of Nova. This object is used when we need to list Nova VM.

Each of these elements can belonged to one or more categories, here is an example of categories:

```
$ moon subject category list
+----------------------------------+------------------------+------------------------+
| id                               | name                   | description            |
+----------------------------------+------------------------+------------------------+
| 3d97b1e12f6949cfa71e6ecd6f15a361 | domain                 | domain                 |
| 366c308036b74c9da9121759a42c2f19 | role                   | role                   |
| f6f7e1fd031144b2a8c4d7866424b8c6 | subject_security_level | subject_security_level |
+----------------------------------+------------------------+------------------------+
$ moon object category list
+----------------------------------+-----------------------+-----------------------+
| id                               | name                  | description           |
+----------------------------------+-----------------------+-----------------------+
| 3a4c9f8e5b404d1db7aa641714c8b1c7 | object_id             | object_id             |
| 292d8f613dea49ec9118f76691e580d1 | object_security_level | object_security_level |
| 57f18e690cb948d88c26d210289fb379 | type                  | type                  |
+----------------------------------+-----------------------+-----------------------+
$ moon action category list
+----------------------------------+-----------------+-----------------+
| id                               | name            | description     |
+----------------------------------+-----------------+-----------------+
| cdfbf00d0c1f4d61bf4b6de669721f10 | access          | access          |
| b01d380dda324e39a6a6b0d09065a93d | resource_action | resource_action |
+----------------------------------+-----------------+-----------------+
```

For example, a subject can have a domain and/or have a specific role and/or have a specific security level (subject_security_level). To know the scope of a category, you can use the *moon scope list <category_id>* command:

```
$ moon subject scope list 366c308036b74c9da9121759a42c2f19
+----------------------------------+-------+-------------+
| id                               | name  | description |
+----------------------------------+-------+-------------+
| 98e0357500274d30bb5ba2f896fbedf9 | dev   | dev         |
| 6e4570266b1f42bab47498714716dca6 | admin | admin       |
+----------------------------------+-------+-------------+
```

In this example, for the category *role*, we have 2 possible values: *dev* and *admin*.

```
$ moon object scope list 292d8f613dea49ec9118f76691e580d1
+----------------------------------+--------+-------------+
| id                               | name   | description |
+----------------------------------+--------+-------------+
| e90edf39b4cc496cb28094a056089d65 | low    | low         |
| 73feffe318de4390bc8b4fce5f7d4b88 | high   | high        |
| 41a80336362248e39298b6f52c4ae14d | medium | medium      |
+----------------------------------+--------+-------------+
```

In this example, for the category *object_security_level*, we have 3 possible values: *low*, *medium* and *high*.

Note: if you try to list a scope with the wrong category ID, MoonClient will raise an error:

```
$ moon subject scope list 292d8f613dea49ec9118f76691e580d1
Getting an error while requiring /moon/intra_extensions/e75dad8f2f7d40de9921b0d444198973/subject_scop
```

Each of these elements (subject, object, action and their respective categories and scopes) can be modified with the sub-commands *add* and *delete*.

To link all of these elements, you can use assignment. In the following example, the subject *admin* is linked to the category *role* with the scope *admin*:

```
$ moon subject assignment list 8101e73fb82e433fbc576587e6201bfe 366c308036b74c9da9121759a42c2f19
+----------------------------------+-------+
| id                               | name  |
+----------------------------------+-------+
| 6e4570266b1f42bab47498714716dca6 | admin |
+----------------------------------+-------+
```

This means that the user *admin* has the role *admin*. In the following example, the subject *admin* is also linked to the category *subject_security_level* with the scope *high*:

```
$ moon subject assignment list 8101e73fb82e433fbc576587e6201bfe f6f7e1fd031144b2a8c4d7866424b8c6
+----------------------------------+------+
| id                               | name |
+----------------------------------+------+
| 45d6852c4a08498298331bcd72c2e988 | high |
+----------------------------------+------+
```

As before, if you put a wrong subject ID or a wrong subject category ID, MoonClient will raise an error:

```
$ moon subject assignment list 8101e73fb82e433fbc576587e6201bfe f6f7e1fd031144b2a8c4d7866424b8c3
Getting an error while requiring /moon/intra_extensions/e75dad8f2f7d40de9921b0d444198973/subject_assi
$ moon subject assignment list 8101e73fb82e433fbc576587e6201bfr f6f7e1fd031144b2a8c4d7866424b8c6
Getting an error while requiring /moon/intra_extensions/e75dad8f2f7d40de9921b0d444198973/subject_assi
```

3.7 Configuration

Before dealing with rules, we must configure our intra_extension. This configuration can be done with *template*, *submetarule*, *aggregation* commands.

We have already see what the *template* command does:

```
$ moon template list
+--------------------+-------------------+-----------------------------+
| id                 | name              | description                 |
+--------------------+-------------------+-----------------------------+
| policy_rbac_admin  | RBAC Admin Policy |                             |
| policy_root        | Root Policy       | root extension              |
| policy_authz       | Multiple_Policy   | Multiple Security Policies  |
| policy_empty_admin | Empty_Policy      | Empty Policy                |
| policy_empty_authz | MLS_Policy        | Multi Level Security Policy |
| policy_mls_authz   | MLS_Policy        | Multi Level Security Policy |
+--------------------+-------------------+-----------------------------+
```

This command only list available policy template. Those templates are hardcoded into Moon, you cannot modify them though the MoonClient. If you need to update them (which is not recommended), you must go in the directory */etc/keystone/policies* and update the json file inside. Those template describe the behaviour of an intra_extension. When you start using Moon, we recommend that you use the *Multiple_Policy* (with ID *policy_authz*) template which is the simplest template. It has default values easy to configure.

This policy template is configured with 3 sub-meta-rules shown below:

```
$ moon submetarule show
+----------------------------------+-----------+-----------+------------------------+-----------------
| id                               | name      | algorithm | subject categories     | object categori
+----------------------------------+-----------+-----------+------------------------+-----------------
| a0c30ab9f4104098a9636b0aab294deb | rbac_rule | inclusion | role, domain           | object_id
| 6e4abecb486448309ad5ace17ab134dc | dte_rule  | inclusion | domain                 | type
| ba9eac79b38a46cc9ab65feb32696803 | mls_rule  | inclusion | subject_security_level | object_security
+----------------------------------+-----------+-----------+------------------------+-----------------
```

Each sub-meta-rules indicates how rules will be built. In this example, the first sub-meta-rules (*rbac_rule*) indicates that a single rule will be the concatenation of the following categories:

- role,
- domain
- object_id
- access

The order between categories is important!

This sub-meta-rules matches a enhanced Role-Base-Access-Control policy. A standard RBAC policy would be:

- role,
- object_id
- access

And we would have in Moon:

```
+----------------------------------+-----------+-----------+------------------------+-----------------
| id                               | name      | algorithm | subject categories     | object categori
+----------------------------------+-----------+-----------+------------------------+-----------------
...
| a0c30ab9f4104098a9636b0aab294deb | rbac_rule | inclusion | role                   | object_id
...
+----------------------------------+-----------+-----------+------------------------+-----------------
```

If you want to modify that point, use the following commands:

```
$ moon subject category list
+----------------------------------+------------------------+------------------------+
| id                               | name                   | description            |
+----------------------------------+------------------------+------------------------+
| 3d97b1e12f6949cfa71e6ecd6f15a361 | domain                 | domain                 |
| 366c308036b74c9da9121759a42c2f19 | role                   | role                   |
| f6f7e1fd031144b2a8c4d7866424b8c6 | subject_security_level | subject_security_level |
+----------------------------------+------------------------+------------------------+
$ moon submetarule set --subject_category_id 366c308036b74c9da9121759a42c2f19 a0c30ab9f4104098a9636b0
$ moon submetarule show
+----------------------------------+-----------+-----------+------------------------+-----------------
| id                               | name      | algorithm | subject categories     | object categori
+----------------------------------+-----------+-----------+------------------------+-----------------
| a0c30ab9f4104098a9636b0aab294deb | rbac_rule | inclusion | role                   | object_id
| 6e4abecb486448309ad5ace17ab134dc | dte_rule  | inclusion | domain                 | type
| ba9eac79b38a46cc9ab65feb32696803 | mls_rule  | inclusion | subject_security_level | object_security
+----------------------------------+-----------+-----------+------------------------+-----------------
```

Warning: After modifying the sub-meta-rule, you **must** delete all rules corresponding to that sub-meta-rule and add new rules (see below).

As you can see, the third column is titled *algorithm*. This algorithm indicates how the match between scopes and rules is done. There are 2 hardcoded algorithms: *inclusion* and *comparison*. At this time the *comparison* algorithm is a future work, don't use it. Use exclusively the *inclusion* algorithm.

3.8 Rules

Rules are analysed by our engine to authorize (or not) an action from Nova or Swift. Here is an example of what a list of rules looks like for the our *rbac_rule* sub-meta-rule:

```
$ moon rule list a0c30ab9f4104098a9636b0aab294deb
+---+----------------------------------+--------+----------+----------+-------------+---------+
|   | id                               | s:role | s:domain | a:access | o:object_id | enabled |
+---+----------------------------------+--------+----------+----------+-------------+---------+
| 0 | b8579d7e2eba4c44a9524843d1b4b2e6 | admin  | xx       | read     | servers     | True    |
| 1 | 11fa000905654737b2476d06fc9e2be0 | admin  | ft       | read     | servers     | True    |
| 2 | 2acca0c356c946d1adec541ad56839ab | dev    | xx       | read     | servers     | True    |
+---+----------------------------------+--------+----------+----------+-------------+---------+
```

In the sub-meta-rule *rbac_rule*, we have 4 categories (role, domain, object_id, access). So we have 4 columns for each rules:

- s:role
- s:domain
- a:access
- o:object_id

The prefix indicates if the category is a subject, action or object category. Here, we have two subject categories, one action category and one object category. Again, the order is very important.

To add a new rule, the help command can be usefull:

```
$ moon help rule add
usage: moon rule add [-h] [--intraextension <intraextension-uuid>]
                     <submetarule-uuid> <argument-list>
```

```
Add a new rule.

positional arguments:
  <submetarule-uuid>    Sub Meta Rule UUID
  <argument-list>       Rule list (example: admin,start,servers) with that
                        ordering: subject, action, object

optional arguments:
  -h, --help            show this help message and exit
  --intraextension <intraextension-uuid>
                        IntraExtension UUID
```

We can see that we need the submetarule-uuid and an argument list. To be more user-friendly, this list uses name of scope and not their ID. You must respect the order : subject scopes, action scopes and object scopes. And if you have more than one scope (in subject for example), you must follow the order of the configuration in the sub-meta-rule. In our example, the order is role then domain.

A new rule will be added like this:

```
$ moon rule add a0c30ab9f4104098a9636b0aab294deb dev,ft,read,servers
$ moon rule list a0c30ab9f4104098a9636b0aab294deb
+---+----------------------------------+--------+----------+----------+-------------+---------+
|   | id                               | s:role | s:domain | a:access | o:object_id | enabled |
+---+----------------------------------+--------+----------+----------+-------------+---------+
| 0 | b8579d7e2eba4c44a9524843d1b4b2e6 | admin  | xx       | read     | servers     | True    |
| 1 | 24ef8de4526f4268a7de530443edd9fa | dev    | ft       | read     | servers     | True    |
| 2 | 11fa000905654737b2476d06fc9e2be0 | admin  | ft       | read     | servers     | True    |
| 3 | 2acca0c356c946d1adec541ad56839ab | dev    | xx       | read     | servers     | True    |
+---+----------------------------------+--------+----------+----------+-------------+---------+
```

The latest column allows to enabled or disabled a specific rule.

3.9 Log system

Logs can be obtain with the *log* command:

```
$ moon log --number 10
+---------------------+--------------------------------------------------------------------------
| Time                | Message
+---------------------+--------------------------------------------------------------------------
| 2016-08-11-12:28:58 | No Intra_Admin_Extension found, authorization granted by default.
|                     |
| 2016-08-12-03:30:05 | /MoonError/AdminException/AdminMetaData/SubjectCategoryUnknown (The given sub
|                     |
| 2016-08-12-03:38:11 | /MoonError/AdminException/AdminPerimeter/SubjectUnknown (The
|                     | given subject is unknown.)
|                     |
| 2016-08-12-03:48:05 | /MoonError/AdminException/AdminMetaData/SubjectCategoryUnknown (The given sub
|                     |
| 2016-08-12-03:48:22 | /MoonError/AdminException/AdminPerimeter/SubjectUnknown (The
|                     | given subject is unknown.)
|                     |
| 2016-08-12-03:50:59 | No Intra_Admin_Extension found, authorization granted by default.
|                     |
| 2016-08-12-03:51:38 | No Intra_Admin_Extension found, authorization granted by default.
|                     |
| 2016-08-12-03:52:01 | No Intra_Admin_Extension found, authorization granted by default.
|                      |
| 2016-08-12-03:52:08  | No Intra_Admin_Extension found, authorization granted by default.
|                      |
| 2016-08-12-03:54:52  | No Intra_Admin_Extension found, authorization granted by default.
|                      |
+----------------------+------------------------------------------------------------------------------
```

In this example, we limit the number of events to 10. You can filter with a particular string or search by date. See the *help* command for more information:

```
$ moon help log
usage: moon log [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
                [--max-width <integer>] [--noindent]
                [--quote {all,minimal,none,nonnumeric}]
                [--filter <filter-str>] [--fromdate <from-date-str>]
                [--todate <to-date-str>] [--number <number-int>]

List all logs.

optional arguments:
  -h, --help            show this help message and exit
  --filter <filter-str>
                        Filter strings (example: "OK" or "authz")
  --fromdate <from-date-str>
                        Filter logs by date (example: "2015-04-15-13:45:20")
  --todate <to-date-str>
                        Filter logs by date (example: "2015-04-15-13:45:20")
  --number <number-int>
                        Show only <number-int> logs

output formatters:
  output formatter options

  -f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}
                        the output format, defaults to table
  -c COLUMN, --column COLUMN
                        specify the column(s) to include, can be repeated

table formatter:
  --max-width <integer>
                        Maximum display width, <1 to disable. You can also use
                        the CLIFF_MAX_TERM_WIDTH environment variable, but the
                        parameter takes precedence.

json formatter:
  --noindent            whether to disable indenting the JSON

CSV Formatter:
  --quote {all,minimal,none,nonnumeric}
                        when to include quotes, defaults to nonnumeric
```

3.10 Test

Moonclient can execute some tests written in a custom format (JSON format). After installing MoonClient, it is advised to execute all tests to see if the Moon platform is up and running:

```
$ moon test  --self
Write tests output to /tmp/moonclient_test_20160812-090856.log

Executing /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_empty_policy_new_user.json (1
...
+------------------------------------------------------------------------------------------+---------
| filename                                                                                 | results
+------------------------------------------------------------------------------------------+---------
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_actions.json               | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_configuration.json         | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_empty_policy_nova.json     | False
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_action_categories.json     | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_action_scopes.json         | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_object_assignments.json    | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_subject_scopes.json        | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_objects.json               | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_subjects.json              | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_object_categories.json     | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_root_intraextensions.json  | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_subject_assignments.json   | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_subject_categories.json    | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_admin_intraextensions.json | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_submetarules.json          | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_object_scopes.json         | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_action_assignments.json    | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_rules.json                 | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_empty_policy_swift.json    | False
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_external_commands.json     | False
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_tenants.json               | True
| /usr/local/lib/python2.7/dist-packages/moonclient/tests/tests_empty_policy_new_user.json | False
+------------------------------------------------------------------------------------------+---------
```

Executing all tests may take time, so be patient. Each test can be executed separately and you have acces to a file log in the */tmp* directory for each test.

Revision:

Build date: August 12, 2016