OpenNebula - Feature #203
Authentication & Authorization Drivers

03/05/2010 12:32 AM - Ruben S. Montero

Status:	Closed	**Start date:**	03/05/2010
Priority:	Normal	**Due date:**	
Assignee:	Ruben S. Montero	**% Done:**	0%
Category:	Core & System	**Estimated time:**	0.00 hour
Target version:	Release 2.0		
Resolution:	fixed	**Pull request:**	

Description

OpenNebula supports multiple users and features basic authentication and authorization policies. This development campaign aims to provide OpenNebula with a generic auth framework, so:

- Users are identified by abstract key/secret tokens. An underlying driver will then interface with the auth back-end (e.g. LDAP / X509 based / PAM / Policikit...) to authenticate the user.

- General Authorization policies can be implemented, for example quotas or allow a user to submit VMs in a given time frame, user groups....

- The current auth framework will be offered as a default

Associated revisions

Revision baca5e4a - 05/27/2010 10:27 PM - Ruben S. Montero

feature #203: Initial version for the Authorization Manager and Drivers

Revision 4cc03e0e - 05/27/2010 10:29 PM - Ruben S. Montero

feature #203: Added the Auth Manager to the SConstruct script

Revision 42161d1a - 05/29/2010 01:42 AM - Ruben S. Montero

feature #203: Syncrhonization functions for the AuthRequest. Added basic tests for the Authentication Manager

Revision 3b259c33 - 05/29/2010 01:43 AM - Ruben S. Montero

feature #203: removed unneeded files

Revision ebedca1f - 05/31/2010 09:37 PM - Ruben S. Montero

feature #203: Authorize test

Revision a707f730 - 07/06/2010 10:35 AM - Ruben S. Montero

feature #203: Prevent race condition when a AuthRequests timeout. Now timeout timers are handled by the AuthManager.

Revision 3388563a - 07/08/2010 01:45 PM - Ruben S. Montero

feature #203: Conforming new protocol spec

Revision b14eefee - 07/08/2010 03:50 PM - Ruben S. Montero

feature #203: Plain authn/authz policies

Revision 76e83df6 - 07/08/2010 04:50 PM - Ruben S. Montero

feature #203: Simplified driver load procedure and getter

Revision 3317695a - 07/08/2010 05:27 PM - Ruben S. Montero

feature #203: Default message for timeouts

Revision 2cc7ff67 - 07/08/2010 05:28 PM - Ruben S. Montero

feature #203: Nebula deamon starts a generic auth manager if defined

Revision 284a2db7 - 07/08/2010 05:29 PM - Ruben S. Montero

feature #203: Authenticate now uses the Auth Manager infrastructure

Revision 04a85b90 - 07/08/2010 05:45 PM - Ruben S. Montero

feature #203: New authorize method for users that use the AuthManager subsystem

Revision 4a733a84 - 07/08/2010 06:21 PM - Ruben S. Montero

feature #203: Make use of AuthManager in vm.action method

Revision bf1816e0 - 07/09/2010 06:14 AM - Ruben S. Montero

feature #203: New authenticate.

Revision c7521469 - 07/09/2010 07:14 AM - Ruben S. Montero

feature #203: Encode object_ids (templates) for CREATE operations

Revision fea2b21e - 07/09/2010 07:26 AM - Ruben S. Montero

feature #203: Fix wrong conflict resolution

Revision 41b676e2 - 07/09/2010 07:39 AM - Ruben S. Montero

feature #203: Tests now works with the AuthManager

Revision f9dbce89 - 07/09/2010 07:42 AM - Ruben S. Montero

feature #203: Restored original scons helpers

Revision edea2edf - 07/09/2010 10:10 AM - Ruben S. Montero

feature #203: Integrated VM allocation with AuthManager

Revision 3fdd16f0 - 07/09/2010 10:43 AM - Ruben S. Montero

feauture #203: Integrated VM operations with AuthManager

Revision 91722661 - 07/09/2010 11:09 AM - Ruben S. Montero

feature #203: Hosts can be used by anybody in plain auth

Revision 1953d814 - 07/09/2010 11:38 AM - Ruben S. Montero

feature #203: Hosts use AuthManager

Revision 0efe68bf - 07/09/2010 05:29 PM - Tino Vázquez

feature #203: Image use AuthManager

Revision f7d3833c - 07/09/2010 05:32 PM - Tino Vázquez

feature #203: Adding authorization for Image allocate

Revision 6641caef - 07/09/2010 05:56 PM - Tino Vázquez

feature #203: Auth for VirtualNetworks

Revision 7777e913 - 07/09/2010 05:56 PM - Tino Vázquez

feature #203: Missing header

Revision aea644f9 - 07/09/2010 06:12 PM - Tino Vázquez

feature #203: Auth for USERS

Revision 9f7544e6 - 07/12/2010 03:45 PM - Ruben S. Montero

feature #203: Cluster methods uses Auth Manager

Revision 2fe5b7b7 - 07/12/2010 04:42 PM - Ruben S. Montero

feature #203: Plain authorize method fix

Revision 89d60d65 - 07/12/2010 05:03 PM - Tino Vázquez

feature #203: Removed unused user info method

Revision 822f2bd0 - 07/12/2010 05:38 PM - Ruben S. Montero

feature #203: Fix compilation error

Revision 8a433f45 - 07/12/2010 05:47 PM - Ruben S. Montero

feature #203: Authorize methods for NET and IMAGE creation in insert

Revision 3a76fea7 - 07/12/2010 06:06 PM - Tino Vázquez

feature #203: Removed authorization from Image and VirtualNetworks in the RM

Revision 16f7982c - 07/12/2010 06:10 PM - Tino Vázquez

feature #203: Added error log functions to RM

Revision a342d00d - 07/12/2010 08:48 PM - Javi Fontan

feature #203: public flag is taken into account

Revision b624debb - 07/13/2010 10:32 AM - Javi Fontan

feature #203: added default and unlimited quota limits

Revision fcb7ac0a - 07/13/2010 11:49 AM - Tino Vázquez

feature #203: Changed RM to use log functions

Revision ccabd550 - 07/13/2010 11:50 AM - Javi Fontan

feature #203: Added tests for simple_permissions and quota setup

Revision 0659115f - 07/13/2010 04:16 PM - Tino Vázquez

feature #203: Finishing RM authorization task

Revision f10b7aff - 07/13/2010 04:17 PM - Javi Fontan

feature #203: added auth mad config file

Revision 213029ad - 07/13/2010 04:18 PM - Javi Fontan

feature #203: simple permissions check for quota

Revision dace9afa - 07/13/2010 04:20 PM - Javi Fontan

feature #203: configuration loading and debug

Revision c322e7c3 - 07/13/2010 04:57 PM - Tino Vázquez

feature #203: Solving newline on some RM log messages

Revision 24ec057a - 07/13/2010 05:26 PM - Tino Vázquez

feature #203: Changing RM log message codes

Revision 9ed94ae7 - 07/14/2010 02:10 PM - Tino Vázquez

feature #203: Readding the cluster to OpenNebula.rb

Revision 1035fe77 - 07/14/2010 02:21 PM - Ruben S. Montero

feature #203: Removed Auth from allocate (VM, IMAGE & NET). Added NETWORK and NETWORK_ID for NICs and IMAGE and IMAGE_ID for DISKS

Revision f19887fa - 07/14/2010 03:28 PM - Tino Vázquez

feature #203: RM log funtions, -2 introduced for (allocate, parse, ...) errors

Revision 34b2ed6c - 07/14/2010 04:11 PM - Ruben S. Montero

feature #203: Authorization method for VMs now in RM

Revision 0dbb3330 - 07/14/2010 05:37 PM - Ruben S. Montero

feature #203: authorization now in RM.*allocate for IMAGEs and NETs

Revision 498102b7 - 07/14/2010 05:49 PM - Ruben S. Montero

feature #203: Better public in Allocate Methods

Revision ee459c44 - 07/15/2010 09:33 AM - Javi Fontan

feature #203: driver is now multithreaded

Revision c5b0932a - 07/15/2010 09:34 AM - Javi Fontan

feature #203: bug fixes and debug messages

Revision f3fc078b - 07/15/2010 10:43 AM - Tino Vázquez

feature #203: Adding comments, fixing pools accesors in RM

Revision d68855a4 - 07/15/2010 02:04 PM - Carlos Martín

feature #192: New tests for ImagePool::disk_attribute and tests fixed for new ImagePool::allocate parameters (#203).

Revision 57e5c959 - 07/15/2010 02:07 PM - Carlos Martín

feature #192: Tests fixed for new VirtualNetworkPool::allocate parameters (#203).

Revision b6178c1c - 07/15/2010 02:08 PM - Carlos Martín

feature #192: Tests fixed for new VirtualMachinePool::allocate parameters (#203).

Revision c2dab091 - 07/15/2010 02:37 PM - Ruben S. Montero

feature #203: Added some more checks to tests and build options

Revision d5d6d717 - 07/15/2010 03:55 PM - Javi Fontan

feature #203: enabled quota for user authorization

Revision 0dc421ea - 07/16/2010 04:48 PM - Javi Fontan

feature #203: Added rsa authentication

Revision a9446ad4 - 07/20/2010 05:30 PM - Javi Fontan

feature #203: Added documentation for ssh authentication

Revision b0b28bf4 - 07/21/2010 03:51 PM - Javi Fontan

feature #203: added authentication driver selection

Revision e972ee8c - 07/21/2010 04:01 PM - Javi Fontan

feature #203: quota can be enabled in configuration file

Revision e615215a - 07/21/2010 04:20 PM - Javi Fontan

feature #203: added help to oneauth command

Revision 25eb2999 - 07/21/2010 04:30 PM - Ruben S. Montero

feature #203: Better auth for deploy and saveas

Revision 5e553fbf - 07/22/2010 02:32 PM - Javi Fontan

feature #203: authorization policies migrated from the core

Revision 2090e4e5 - 03/01/2017 04:12 PM - Juan Jose Montiel Cano

Master bis (#203)

- removed bug of creation of template and vm
- F #2347 added section for vmgroup into cloud.yaml
- F #2347 added section for select vmgroup when you intanciate a machine
- F #2347 Filling mvgroup when instantiating a vm from a template

History

#1 - 06/23/2010 10:05 PM - Gyula Csom

Hi!

Do you have any plans to update the user domain as well? like introducing additional user attributes, role and organization entities, etc.? Reason behind:

In the short term in our cloud system 1. we must support some additional user attributes like public SSH key (in order to contextualize images), SMI account (which is the account to log into the storage subsystem) and 2. we should also support organizational data. This somehow overlaps the "General Authorization policies" above, but it doesn't seem to be the same feature.

We have some choices (for instance extending the ONE core (including the db and the API as well), implementing some directory integration), however we want to adhere to the ONE mainline as much as possible, since we want a maintanable solution.

So do you have any plan to support such domain extensions either directly in the ONE database or through directory integration/driver? Is this on your roadmap? If not what do you advice what direction should we go onto (extending the core? implementing directory integration?)? What would be the closest approach to your concept?

Cheers,
Gyula

#2 - 06/25/2010 12:48 AM - Ruben S. Montero

Hi,

The reason behind the use of the driver is to include all the extensions without the need to modify the core. The additional information could be stored in a DB maintained by the driver (or any other system like LDAP). Our approach is to keep the user table as slim as possible in the core (i.e. just storing the attributes needed to authenticate with any external system). In this way "password" could be a ssh key or a x509 certificate, and the username can be used by the driver to get additional information (like groups, acls or another tokens like the SMI account)

In order to integrate with any authentication system, the session parameter sent through the XML-RPC interface should also be modified to match the input expected by the driver. We plan to include a simple auth mechanism to show this.
Note that the core will issue authorization and authentication request to the driver, e.g.

- authentication using the session parameter of the XML-RPC and the username and passowrd in the DB
- authorization using a set of actions over OpenNebula objects, like "user wants to create a VM and wants to use vnet 2 and wants to use image 34..." The core will expect a Yes/No answer.

If you want to include additional attributes in the context, we'll have two options:
1.- Add support for including $USER like attributes (like $NETWORK, that has been recently implemented) and will only give you access to username and password - This is not planned for 1.6
2.- Make tm_context.sh to talk to your auth driver to put in the iso the additional data.

Feedback is more than welcome

Cheers

Ruben

#3 - 06/25/2010 08:20 AM - Gyula Csom

Hi!

Thank you for your response! I think I've got the point: the idea is to outsource auth/user related datas/functionalities to an auth/directory backend and keep the core thin. Regarding our use cases this means:

1. Contextualize vms by the user's SSH key: then we might use the second option and get the key through the driver.
2. Log into the storage with the user's credentials in order to dinamically create/delete iSCSI target/LUN: then again we can use the second option.
3. Manage organizational data: this would be outsourced to the backend.

All that the above functions need is the vm id in order to determine the user of it. This goes back to an earlier discussion (http://dev.opennebula.org/issues/216#note-10).

For now my only question is the following:

Will this new auth logic be applied to other elements, too? For instance when the user wants to use a virtual network will the auth (driver) intercept such calls and accept/deny the request based on auth rules?

Cheers,
Gyula

#4 - 06/28/2010 12:57 AM - Ruben S. Montero

Will this new auth logic be applied to other elements, too? For instance when the user wants to use a virtual network will the auth (driver) intercept such calls and accept/deny the request based on auth rules?

Yes,

- Networks, apart from authorize the request (i.e. getting a lease from a network) you could also set user quotas...
- Hosts, again the idea would be (user X wants to deploy VM y on host z) the authorization will include the host so you could set host acl's (e.g. in a hybrid setup only certain users are allowed to outsource resources to an external cloud)
- This will also work for the new Image Pool

I've thinking about the context issue... We could add just a new column to our current username/password table for the user pool. If we add a context column the SSH keys could be stored there. The context column would store "opaque" data that would be interpreted by the contextualization scripts inside the VM...

In this way a user would be defined (at the core level) by an username, a security token and a context data. What do you think?

#5 - 06/28/2010 07:02 PM - Gyula Csom

What do you think?
I think adding a custom, opaque field to the user entity is a simple way to make it extensible. Also this might be necessary if someone wants a compact system (ie. she doesn't want to maintain an auth backend) but wants some customization. Perhaps the idea can work for other cases (like images) as well.

Meanwhile I've discussed the topic with our customer's developer and so far we came to the conclusion that we would try the centralized approach that would be inline with this feature, the original intentions. We will factor out auth logic from our subsystems and put it into an auth subsystem shared/used by the others. The interesting thing here is that we will try an active, "SSO-like" system that supports directory services and auth as well.

This is our preliminary idea, which (I guess) is inline with the 1.6 intentions. Currently we think directory data and authentication is not an issue, however authorization is challenging. That is because authz intermixes domain specific thing (methods, resources, etc. that is being authenticated) and

the authz logic which is generic. We are currently thinking about a token based (or "one time ticket") approach, that is the auth system will issue one time access tickets for individual request-response cycles that is requested by the clients (here the cloud) and validated by the servers (for instance the storage subsystem).

This is our general idea, now we "just" have to look after the platform that supports this model or kinda:) The customer currently uses Shibboleth but it seems to support just web-only requests. Anyway if we made some progress I would feedback here... if you think it might be useful for you as well. Any opinion?:)

Cheers,
Gyula

#6 - 06/28/2010 09:55 PM - Ruben S. Montero

Let us put the user-based contextualization in stand-by, as this can be achieved by other (not so glamorous) means. I'll be definitely interested in your solution and any other feedback you have to make the auth subsystem more useful. Also this would be very interesting for other people integrating the authentication solution of their datacenter.

Thanks for your feedback

Ruben

#7 - 06/29/2010 03:31 PM - Ruben S. Montero

Some details on the communication protocol between the core and the Authorization/Authentication Manager.

The Core will issue the following requests

- The core will send the following string to authenticate a user

  AUTHENTICATE <REQUEST_ID> <USER_ID> <USER_NAME> <PASSWORD> <XML_RPC TOKEN>

  where
  - REQUEST_ID is an unique ID for the operation, should be used by the driver to communicate the result back to the core
  - USER_ID, USER_NAME and PASSWORD are the user info as stored in the DB
  - XML_RPC TOKEN is the xmlrpc session parameter

The session is a cryptographic challenge that should be check using the user_id, user_name and password (and possibly any other third- party authentication system).

The driver response is

  AUTHENTICATE <SUCCESS|FAILURE> <REQUEST_ID> <MESSAGE>

- The core will send the following string to authorize a user request

  AUTHORIZE <REQUEST_ID> <USER_ID> <AUTHORIZATION_TOKEN_1> ... <AUTHORIZATION_TOKEN_N>

where AUTHORIZATION_TOKEN_1 is an authorization request over a given OBJECT (network, image, virtual image and host) in the form:

OBJECT:OBJECT_ID:ACTION:OWNER:PUBLIC

- OBJECT = {VNET, VM, IMAGE, HOST}
- ACTION = {CREATE,USE,MANAGE,DELETE}
- PUBLIC = {0,1}

In this way "user 2 wants to create a VM that uses network red and image ubuntu" will translate to the authorization string

VM:-:CREATE:2:- NET:2:USE:0:1 IMAGE:4:USE:2:0

The driver should answer yes/no (SUCCESS/FAILURE) if all authentication tokens are granted.

Comments are welcome

Cheers

#8 - 06/29/2010 04:52 PM - Gyula Csom

1. I've stopped at the single line AUTHORIZE request for a 'moment'. Since it may represent a bunch of actions on many entities, first it was not obvious how it will map relations between these entites. Then I concluded that it doesn't have to present these relations at all: the authz system doesn't have to know the 'whys' (why she wants to use network red and image ubuntu) just the 'whats'. Am I right?:)

2. *USER_ID, USER_NAME and PASSWORD are the user info as stored in the DB*
Does this mean the ONE db? Is it really the password not the password hash?

3. *The session is a cryptographic challenge...*
Where does the XML-PRC token data come from? Is it passed by the client with the request? See the next also...

4. How will the new auth model effect the front end (ie. OCA, XML-RPC)?

Below the fold:)

The following might or might not directly effect the ONE core. However in either case it effects implementors and the cloud system as a whole:

5. As I mentioned earlier we are dealing with a distributed system where UI, storage, cloud and directory might be standalone subsystems. We've just started to look after such distributed directory/authn/authz platforms that would serve an architecture like that. The preliminary search didn't give back promising results. That is though there are many OpenSource SSO solutions (CAS, OpenSSO, JOSSO, etc.) and technologies (SAML, OpenID) none of them deals with system integration they are about just web browsers and users not applicable to this field (eg. we don't deal with user agents and long user sessions).

6. So far it seems that building a distributed auth system that is (a) highly available (no single point of failure, subsystems can run even on auth subsystem failure), is (b) fast and scalable (auht backend is not a bottleneck), is (c) consistent (no stale data) and is (d) secure, is a challenge and even it might be impossible (ie. CAP theorem [1]). To my understanding this is a general issue effecting both authentication and authorization.

Cheers

---

[1] CAP theorem: http://en.wikipedia.org/wiki/CAP_theorem

#9 - 06/29/2010 05:00 PM - Gyula Csom

Ops. One more question:

7. Do you plan to support password updates?

#10 - 06/29/2010 06:44 PM - Ruben S. Montero

Gyula Csom wrote:

> *1. I've stopped at the single line AUTHORIZE request for a 'moment'. Since it may represent a bunch of actions on many entities, first it was not obvious how it will map relations between these entites. Then I concluded that it doesn't have to present these relations at all: the authz system doesn't have to know the 'whys' (why she wants to use network red and image ubuntu) just the 'whats'. Am I right?:)*

That's our view also

> *2. USER_ID, USER_NAME and PASSWORD are the user info as stored in the DB*
> Does this mean the ONE db? Is it really the password not the password hash?

It means whatever you put in the password field. The default is a hashed password, but it could be your public ssh key, or a x509 certificate. The underlying driver has to add the semantics to the password field

> *3. The session is a cryptographic challenge...*
> Where does the XML-PRC token data come from? Is it passed by the client with the request? See the next also...
>
> 4. How will the new auth model effect the front end (ie. OCA, XML-RPC)?

Yes, you got it. We plan to support a plug-able module to generate the session token (the current session parameter of every XML-RPC call). The default module is the current username:sha1_password. As an example we plan to include a AWS like session string (i.e. a canonical string signed with the password, e.g. ssl encrypted with private keys). The driver should decrypt session string (using the "password" in the db) and check the canonical string.

> *Below the fold:)*
>
> *The following might or might not directly effect the ONE core. However in either case it effects implementors and the cloud system as a whole:*

We hope that the authn&authz subsytem will make life easier to implementors and cloud architects...

#11 - 07/01/2010 07:12 PM - Javi Fontan

- To enable quota checking in authentication/authorization drivers VM template should be provided as this information is only available in opennebula core. The way to do that is to add the template in the creation token as object_id. It should be encoded as base64 so the message is valid:

VM:dGhpcyBpcyBhIHNhbXBsZSB0ZW1wbGF0ZQ==:CREATE:2:-

#12 - 07/09/2010 05:37 PM - Javi Fontan

For ssh key authentication public key is needed in a different format as ssh format is not supported by openssl libraries. The way to extract rsa public key is:

```
$ openssl rsa -pubout < ~/.ssh/id_rsa
```

for dsa:

```
$ openssl dsa -pubout < ~/.ssh/id_dsa
```

#13 - 09/24/2010 12:47 PM - Ruben S. Montero

- Status changed from New to Closed

- Resolution set to fixed