RED HAT SUMMIT

10 YEARS *and counting*

SAN FRANCISCO | APRIL 14-17, 2014

Red Hat Summit 2014

Red Hat Enterprise Linux Identity Management

Thursday, April 17th 2014

Diaa Radwan

redhat.

Table of Contents

Lab Overview

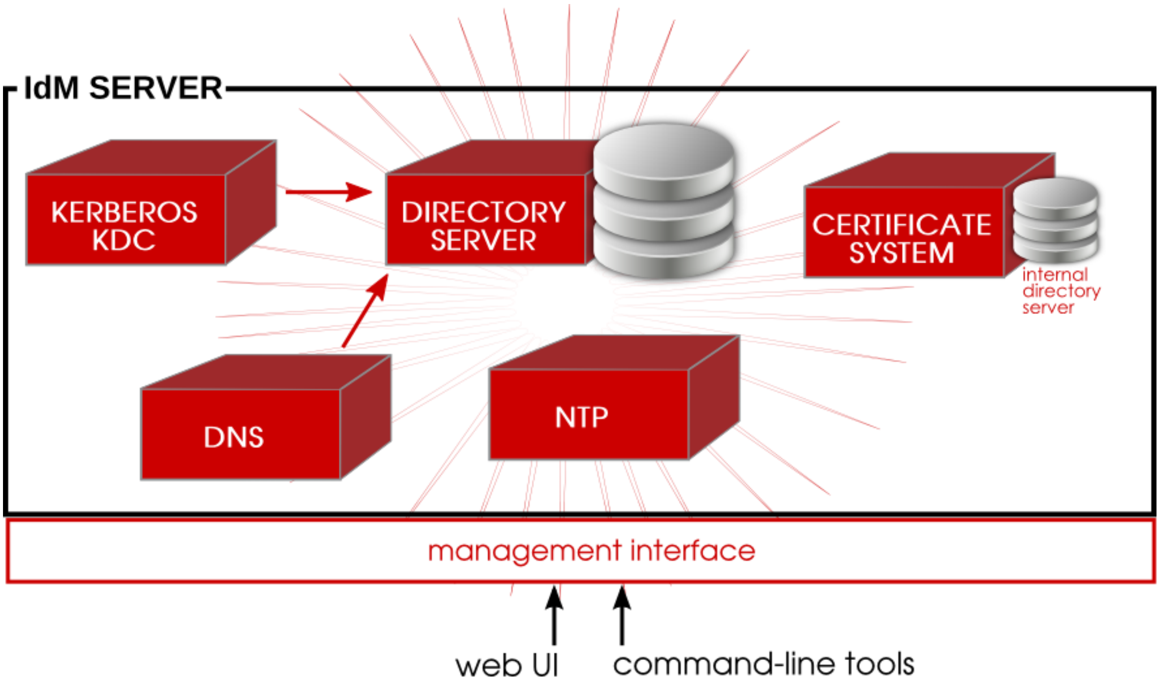
This lab guide assumes that you're following instructor-led training and that this lab guide is will try to simulate real life tasks and scenarios. It goes through a number of labs that will enable your to create full functional environment using Red Hat Enterprise Linux IdM. Also you will explore IdM features such as users, groups, policies and access control rules management. The purpose is to give you a basic hands-on overview of Red Hat Enterprise Linux Identity Management and how the components are fit together. It will use a combination of command-line tools and the IdM web interface. This lab is prepared to run on environment, the setup is descried in this document on Lab Environment Section.

Your instructor will provide you with any additional information that you will require, primarily the lab setup and required scenarios.

Background

Red Hat Enterprise Linux Identity Management Overview

Red Hat Enterprise Linux IdM is a way to create identity stores, centralized authentication, domain control for Kerberos and DNS services, and authorization policies — all on Linux systems, using native Linux tools. It is also supports Linux/Unix domains.



Red Hat Enterprise Linux Identity Management Benefits:

Enhances Security

Centralizes authentication, authorization and fine-grained access control for UNIX/Linux environments.

Provides eSSO (enterprise Single Sign-on)

Enables users to access many different enterprise resources after their initial log-in without having to type user name and password again and again.

Centralizes Administration and Control

Allows administrators to easily consolidate and manage identity servers in a UNIX/Linux environment; with the option to interoperate with Active Directory.

Implements Standards-Based, Integrated Components

Integrates the capabilities of Kerberos, LDAP, DNS and x.509 certificates into a simple identity management solution.

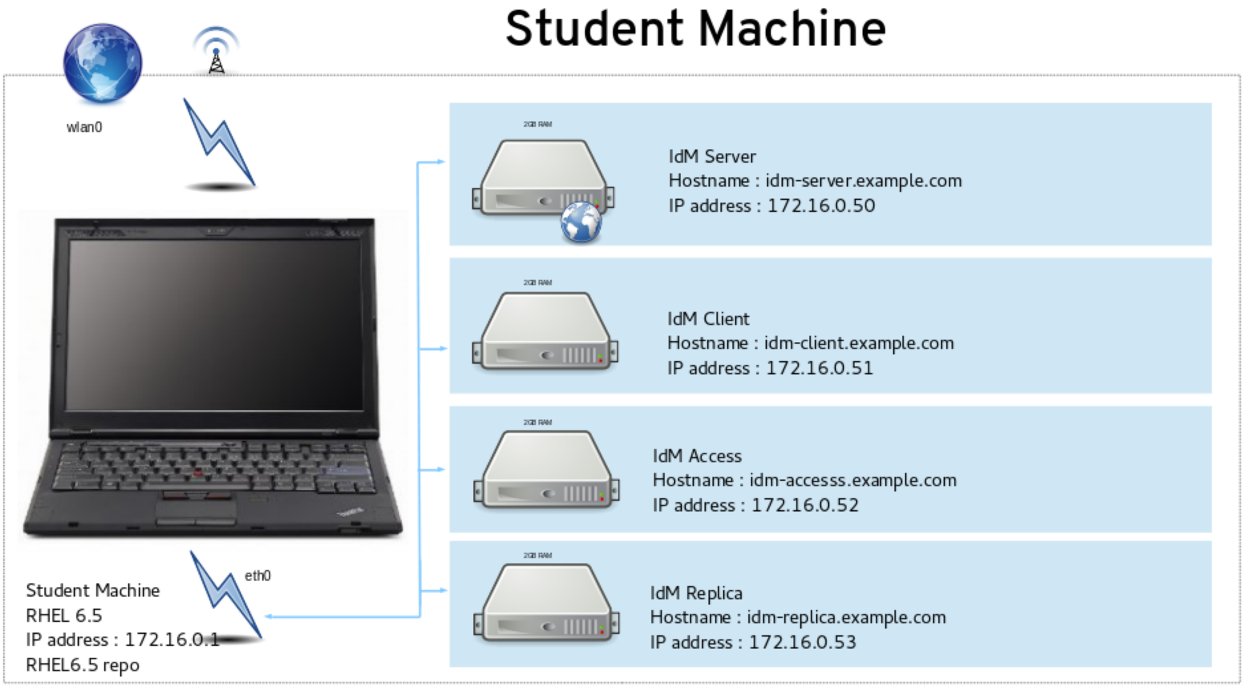
Reduces costs

Can replace third party user directories; and if joined with Linux for UNIX/Linux users identity management, can replace Active Directory, saving on client access licenses costs.

IdM Features

- Integrated, native user, host, and service authentication and access control.
- Consistent and manageable identity management for Linux and Unix systems.
- Interoperability with Microsoft Active Directory domains.
- Standards-based, trusted technologies.
- Easier and clearer to implement, maintain, and understand authentication and access control policies.
- Flexible access control rules based on sudo rules, host-based rules, and other criteria.
- Consistent and universal password policies for users.
- Integrate established Linux/Unix services like NFS, automount, NIS, NTP, Kerberos, and DNS into a single management domain.
- Smooth migration paths from NIS and LDAP services.
- Scalable operations with up to 20 servers and replicas and an unlimited number of clients in a single domain.

IdM Lab Environment Details



Element	URL	Username	Password
IdM Server	http://idm-server.example.com	admin	password
IdM Server	ssh: idm-server.example.com	root	redhat
IdM client	ssh: idm-client.example.com	root	redhat
IdM access evaluation	ssh: idm-access.example.com	root	redhat
IdM Replication	ssh idm-replica.example.com	root	redhat

IdM Lab objectives

Deploy both client and server centralized and high available authentication using Red Hat Enterprise Linux Identity Management (IdM) and provide a working central authentication server, implement additional access controls and sudo rules for client and access machines.

Note: Make sure that all virtual machines starting with “lab13-idm” are running.

Lab 1: Server Installation

Target server: idm-server.example.com
Access: ssh root@idm-server.example.com

- Log into idm-server.example.com, via ssh.
- Make sure that hosts file is properly configured, you should find this line:

```
cat /etc/hosts
172.16.0.50     idm-server.example.com    idm-server
```

- Install the IdM packages:

```
yum -y install bind-dyndb-ldap ipa-server
```

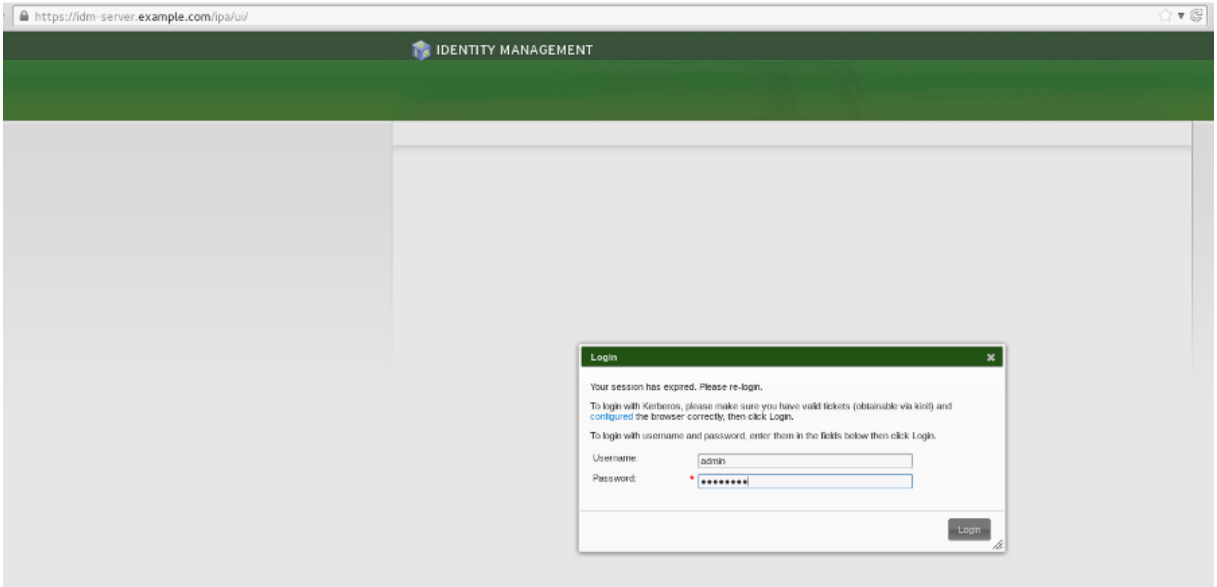
- Run as root:

```
[root@idm-server ~]# ipa-server-install --setup-dns --ssh-trust-dns
```

You should get the same information at end of the dialog:

```
The IPA Master Server will be configured with:
Hostname:      idm-server.example.com
IP address:    172.16.0.50
Domain name:   example.com
Realm name:    EXAMPLE.COM
BIND DNS server will be configured to serve IPA domain with:
Forwarders:    8.8.8.8
Reverse zone:  0.16.172.in-addr.arpa.
```

After installation: Check the IdM web interface via idm-server.example.com.



Your session has expired. Please re-login.

To login with Kerberos, please make sure you have valid tickets (obtainable via kinit) and configured the browser correctly, then click Login.

To login with username and password, enter them in the fields below then click Login.

Username: admin
Password: * ••••••••

- Check main IPA configuration: /etc/ipa/default.conf base DN, realm.
- Obtain a kerberos ticket:

```
kinit admin
klist
```

- Check automatically created DNS records (A, SRV):

```
ipa dnszone-find
ipa dnsrecord-find
```

- Check IdM server defaults:

```
ipa config-show
ipa config-mod --defaultshell=/bin/bash
```

- Then on the idm-server check the logs (Just to know where to start debugging, not needed):

```
/var/log/pki-ca/debug
/var/log/pki-ca-install.log
/var/log/dirsrv/ (permissions!)
/var/log/messages
```

- Common install issues:
  - Broken DNS, bad /ect/hosts configuration.
  - Files and certificates remains after the last unsuccessful install.
  - Time synchronization issues.

Lab 2: Users and Password Policies

Target server: idm-server.example.com
Access: ssh root@idm-server.example.com

- Add new users (create a username with your preferences in the prompt mode):

```
ipa user-add
ipa user-add --first=John --last=Smith jsmith
ipa user-add --first=Matt --last=Well --manager=jsmith
--email=mwell@example.com --homedir=/home/mwell  mwell
```

- Modify User attributes:

```
ipa user-mod jsmith –addattr=departmentnumber=101
ipa user-show jsmith  --all

ipa user-mod mwell --title="System Engineer"
```

- Modify Users password as admin:

  ```
  ipa user-mod mwell --password
  ipa user-mod jsmith --password
  ```

- Check if the system recognize the users:

  ```
  id jsmith
  getent groups mwell
  ```

- Check the default Password Polices:

  ```
  ipa help pwpolicy
  ipa pwpolicy-show
  ipa pwpolicy-mod --maxlife=60
  ```

  - As jsmith login via ssh to idm-server, you will be prompted to change the password for first time. Then Change password with:

    ```
    ipa passwd
    ```

    it will fail because of min life policy.

  - As Admin:

    ```
    ipa pwpolicy-mod --minlife=0 --maxfail=3
    ipa pwpolicy-show
    ```

  - As mwell, login to the idm-server, change 1st time password and then, Change password with *ipa passwd* , it will successed.

- On the Web UI check the following:
  - Add a user.
  - Check password expiry.
  - Edit user details.

Reference:
https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Identity_Management_Guide/managing-users.html

Lab 3: Client Installation

Target server: idm-client.example.com and idm-access.example.com
Access: ssh root@idm-server.example.com

- Check in both servers resolv.conf point to IPA server:

  ```
  cat /etc/resolv.conf
  nameserver 172.16.0.50
  dig example.com | grep ^example.com
  example.com. 3600 IN SOA idm-server.example.com. hostmaster.example.com.
  1396857706 3600 900 1209600 3600
  ```

- Install the IdM client (sssd):

  ```
  yum install ipa-client
  ```

- on IdM server, make sure that PRT records are created/updated in new client installations:

  ```
  ipa dnszone-mod --allow-sync-ptr=TRUE
  ```

- Check and flush iptables rules if there are any rules:

  ```
  iptables -nvL
  iptables -F ; iptables -X
  ```

- Client installation:

  ```
  ipa-client-install --enable-dns-updates --mkhomedir --ssh-trust-dns
  ```

- Additional options to automate the installation:

  ```
  ipa-client-install --mkhomedir --ssh-trust-dns
  --server=idm-server.example.com --domain=example.com -p admin -w
  'password' --fixed-primary -U
  ```

- Some adjustment.
  The default shell for new users is /bin/sh, which should probably be changed:

  ```
  ipa config-mod --defaultshell=/bin/bash
  ```

- Perform the same steps on idm-access.example.com
- Try to access both machines with the above created users from idm-client.example.com.

```
ssh jsmith@idm-access.example.com
Creating home directory for jsmith.
```

- Ssh back to idm-client.example.com, you should login without any passwords:

```
ssh jsmith@idm-client.example.com
```

Note: make sure that you have domainname and hostname in your hosts file. Example: in idm-access.example.com:

```
 172.16.0.51 idm-access.example.com idm-access
```

Lab 4: User Groups and Host Groups Management

Target server: idm-server.example.com
Access: ssh root@idm-server.example.com

Activities for lab 4:

- Create users group (Either through command line or Web UI).
- Adding Group Members.
- Deleting users group.
- Explore IdM group management through command line:

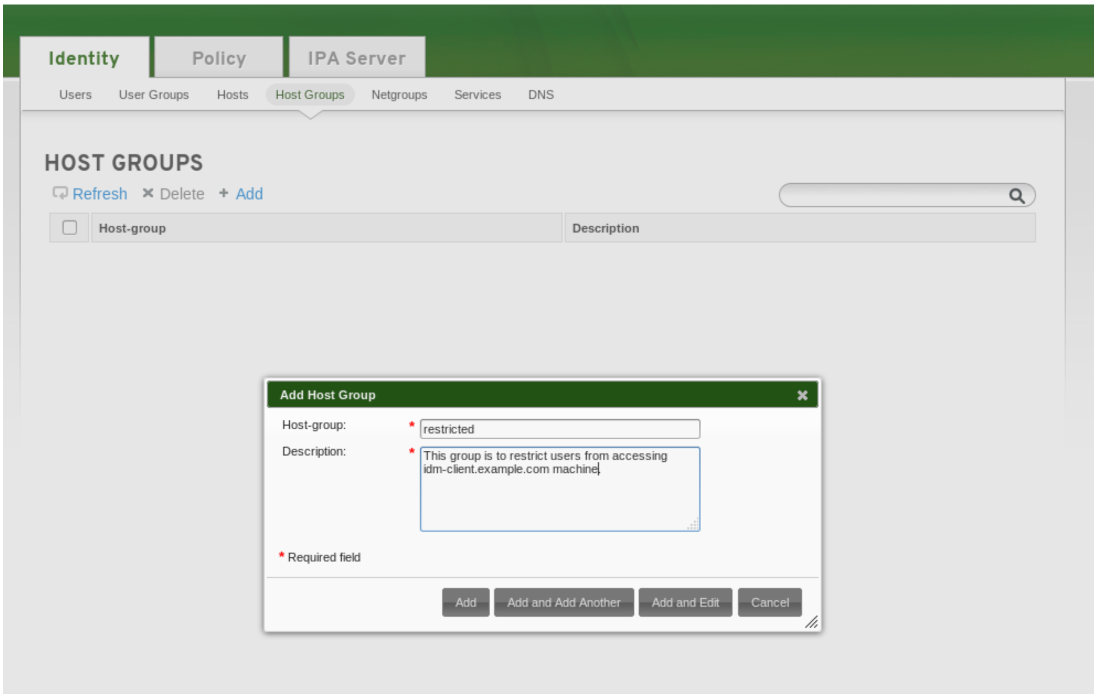
```
ipa help group
ipa group-add --desc='users server group' servers
ipa group-add-member servers --users=mwell
ipa group-add --desc='users client group' clients
ipa group-add-member clients --users=jsmith
ipa group-find
ipa group-del <group name>
```

On the Web UI check the following:

- The group that you just created through the command line.
- Default User and Groups Settings, 3 default groups:
  - ipausers.
  - admins.
  - editors.

- Check the created groups on the web interface, check also the created users.
- Create two host groups:
  - restricted.
  - access.

Through the web interface access follow Identity -> Host groups -> Add.



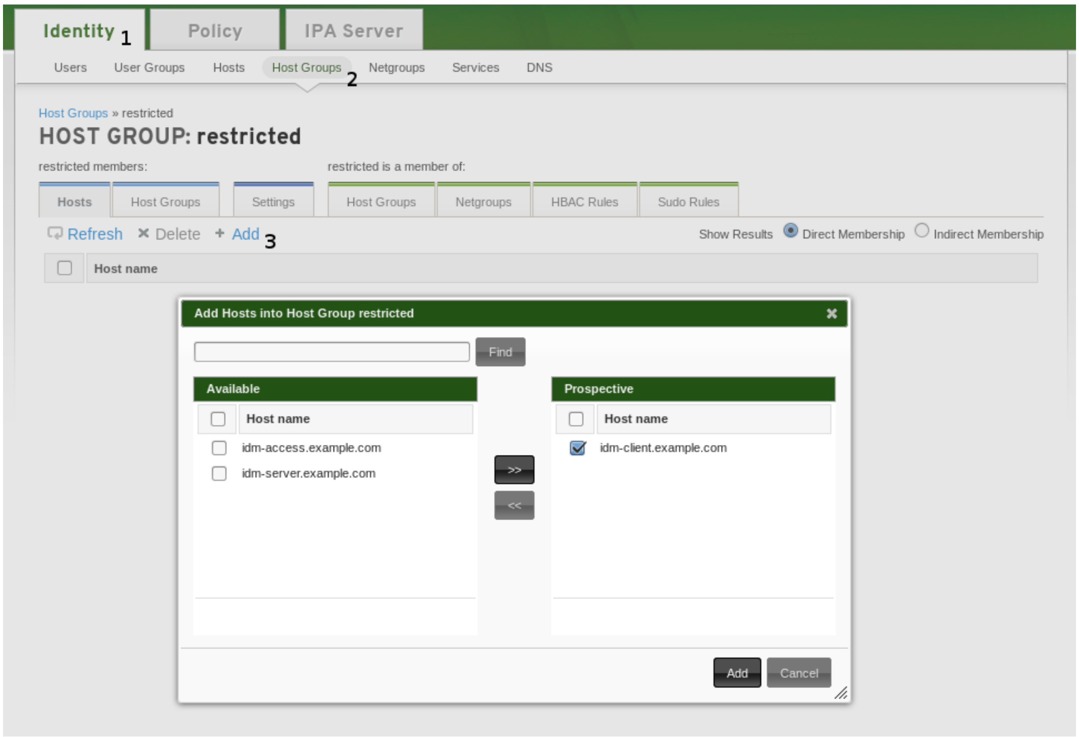
Host-group: * restricted

Description: * This group is to restrict users from accessing idm-client.example.com machine

* Required field

Then you will find a created host group named *'restricted'*, click on the *'restricted'* -> *'Add'*.

- Both idm-client and idm-server should be in restricted group.



HOST GROUP: restricted

Show Results ● Direct Membership ○ Indirect Membership

□ Host name

Add Hosts into Host Group restricted ✕

Available

□ Host name
□ idm-access.example.com
□ idm-server.example.com

Prospective

□ Host name
☑ idm-client.example.com

- Create other host group following the same steps, name this group *'access'*.
- The idm-access machines should be in *'access'* group, follow the same steps in creating and adding machines to restricted group.

Reference:
https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Identity_Management_Guide/user-groups.html

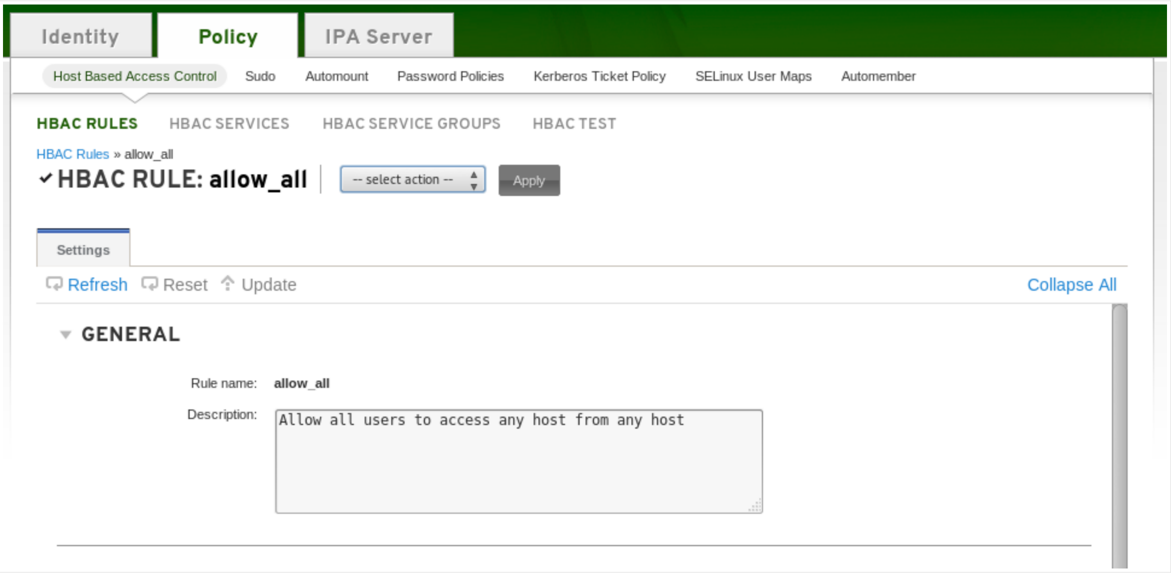
Lab 5: Host Based Access Control – HBAC

Target server: idm-server.example.com, idm-client.example.com and idm-access.example.com

Access: ssh root@idm-server.example.com, ssh root@idm-client.example.com and ssh root@idm-client.example.com

In this Lab we will restrict/allow access based on host groups that we defined in the previous labs. By default IdM is having allow access permission, we could disable it during the installation time through --no_hbac_allow.

- Disable the default allow_all rule through web interface.

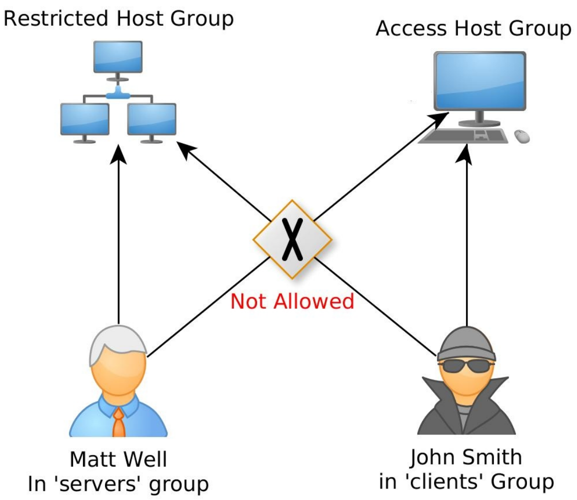


Rule name: allow_all

Description: Allow all users to access any host from any host

We want to grant access permissions to users in *'servers'* group to access all machines considering the following:

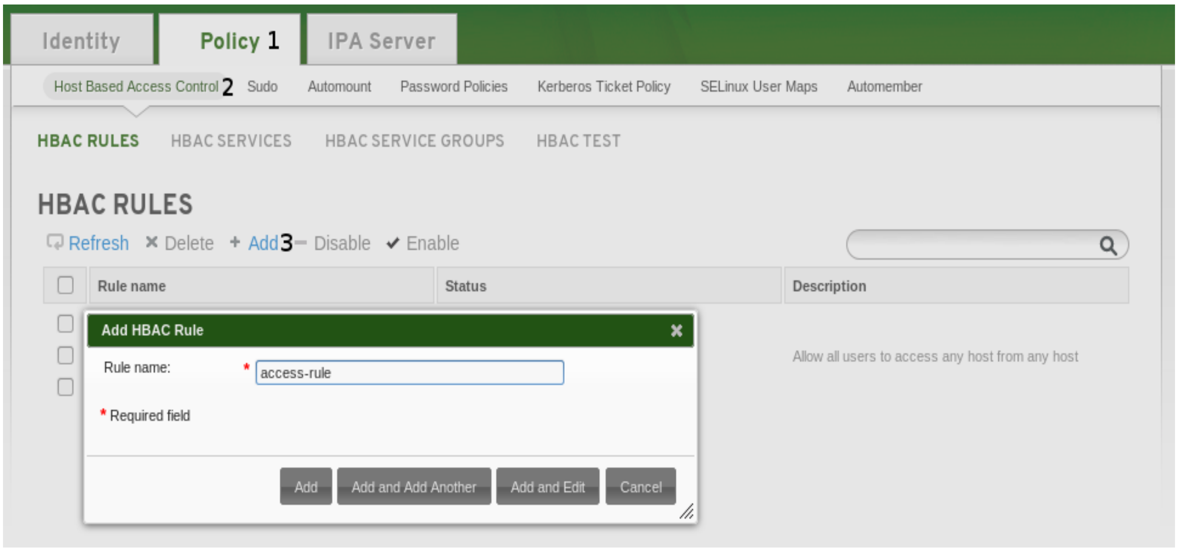
- Users in 'servers' group can access 'restricted' host group servers.
- Users in client group can access Allow host groups only.



The HBAC defines who can access which resources within the environment, not the level of access. This is called host-based access control because the rule defines what hosts are allowed to access other hosts within the domain.

Four basic elements to construct HBAC rule:

- Who: The rule applies to.
- Where: Hosts users can access.
- How: What login services can be accessed.
- Setting Host-Based Access control Rules.
  - HBAC Rule with name *'access-rule'* through the web interface.



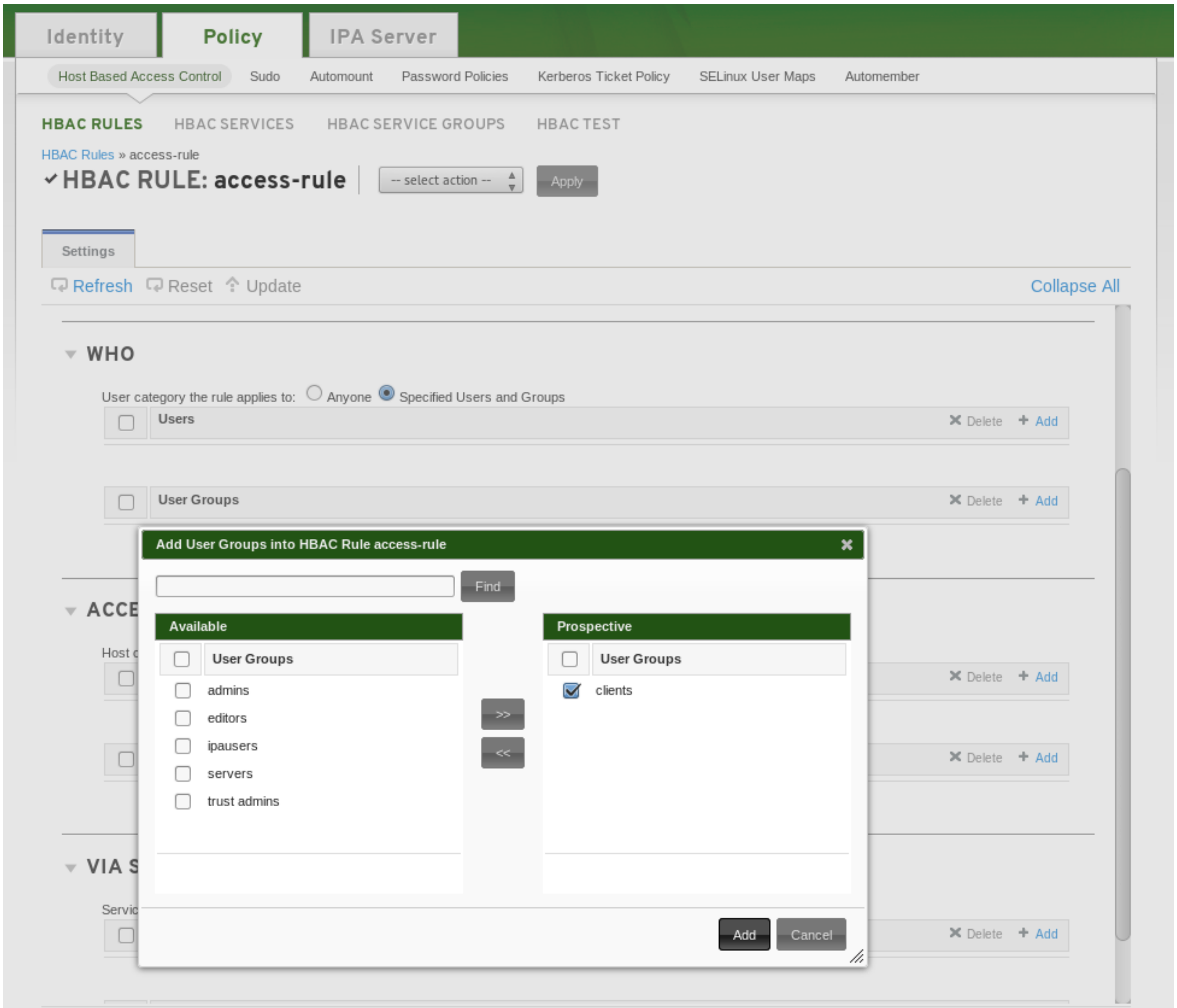
Rule name: * access-rule

* Required field

Allow all users to access any host from any host

  - Click on the access-rule HBAC and add users or users groups this rule will be applied on.

- Add *'clients'* users group to the access-rule in WHO field.



User category the rule applies to: ○ Anyone ● Specified Users and Groups

□ Users

□ User Groups

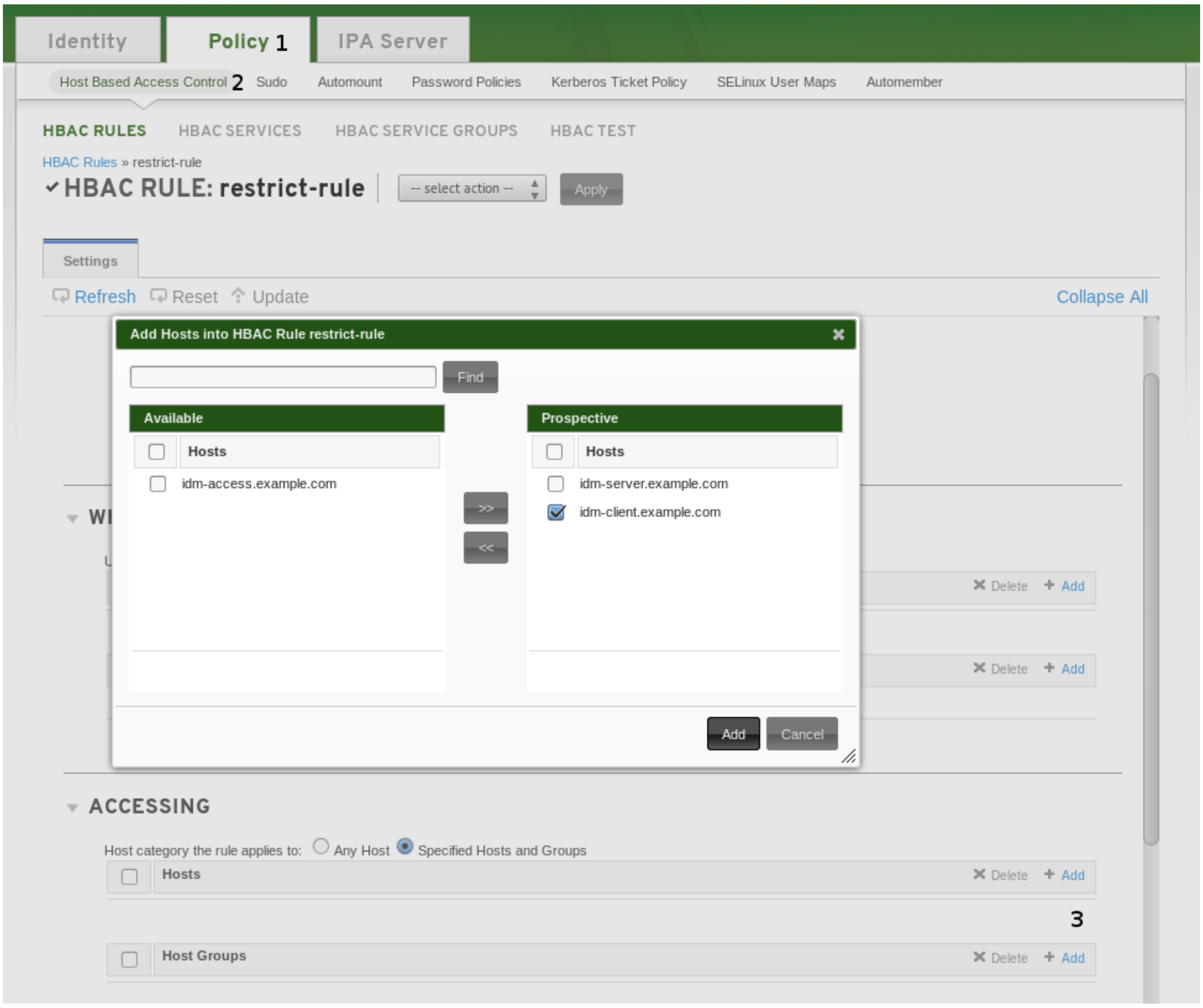
Available:

□ User Groups
□ admins
□ editors
□ ipausers
□ servers
□ trust admins

Prospective:

□ User Groups
☑ clients

  - Then Add the resources that will have these rules applied either host groups or specific hosts (to access-rule).



HBAC Rules » restrict-rule

✓HBAC RULE: restrict-rule

Add Hosts into HBAC Rule restrict-rule

Available
□ Hosts
□ idm-access.example.com

Prospective
□ Hosts
□ idm-server.example.com
☑ idm-client.example.com

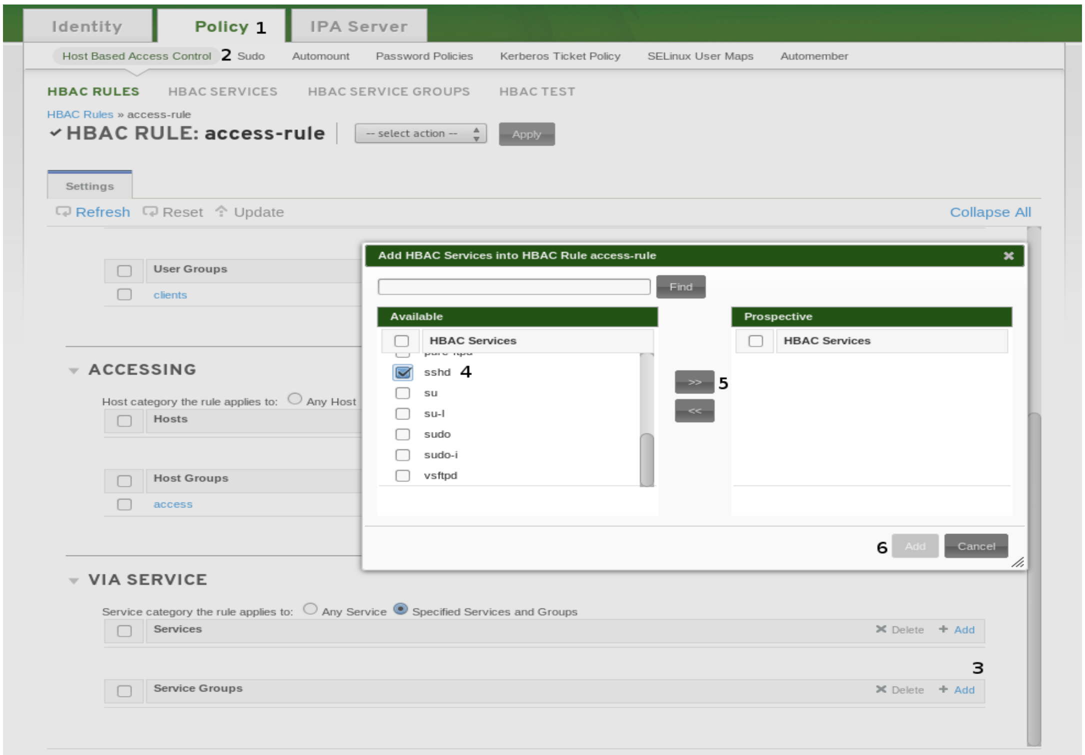
ACCESSING

Host category the rule applies to: ○ Any Host ● Specified Hosts and Groups

□ Hosts

□ Host Groups

- ○ Now we want to add the service that will be allowed, select the sshd and login services:



- In previous steps we added the access-rule that will allow *'clients'* users group to access *'Access'* host groups or the idm-access.example.com.
- Create other HBAC that allows *'servers'* users group to access *'restricted'* host group using the previous example.
- Testing Host-Based Access control Rules:
  - ○ User mwell can ssh to idm-client.example.com successful.
  - ○ User mwell will find access denied message if tried to ssh to *'access'*.
  - ○ User jsmith can login via ssh to access.example.com.

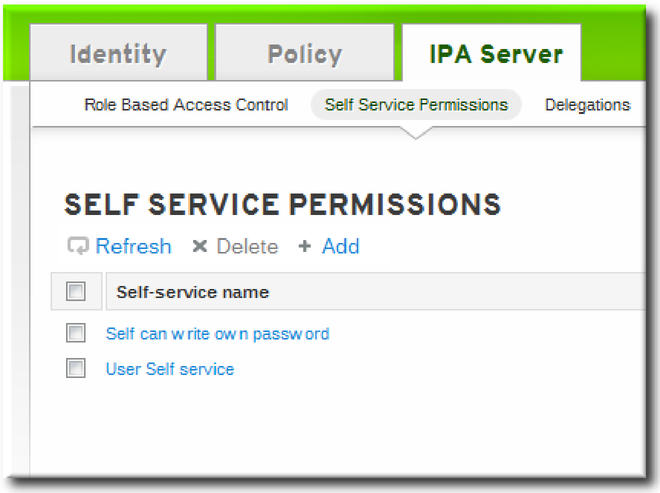
Reference:
https://access.redhat.com/knowledge/docs/enUS/Red_Hat_Enterprise_Linux/6/html/Identity_Management_Guide/configuring-host-access.html

Lab 6: IdM Roles Management

IdM Role Management provides rights or permissions that users have been granted to perform operations within IdM on other users or objects:

- Who can perform the operation.
- What can be accessed.
- What type of operation can be performed.
- Existing Predefined Roles.

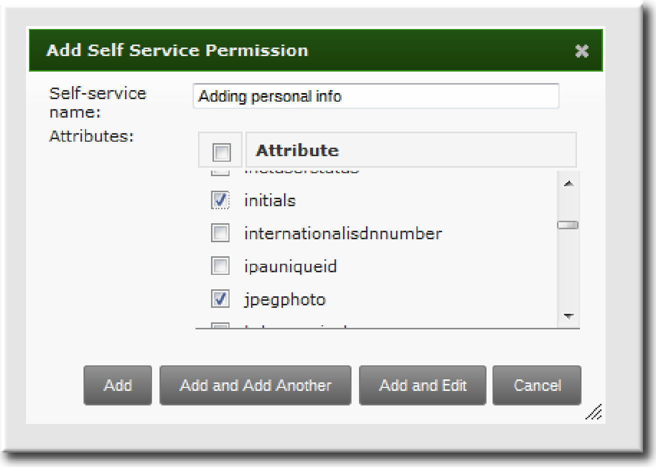
In this lab we will provide privileges to mwell or his group to change his/their user information (e.g car license):



SELF SERVICE PERMISSIONS

□	Self-service name
□	Self can write own password
□	User Self service

Adding the self services permission, Click on *'Add''*:

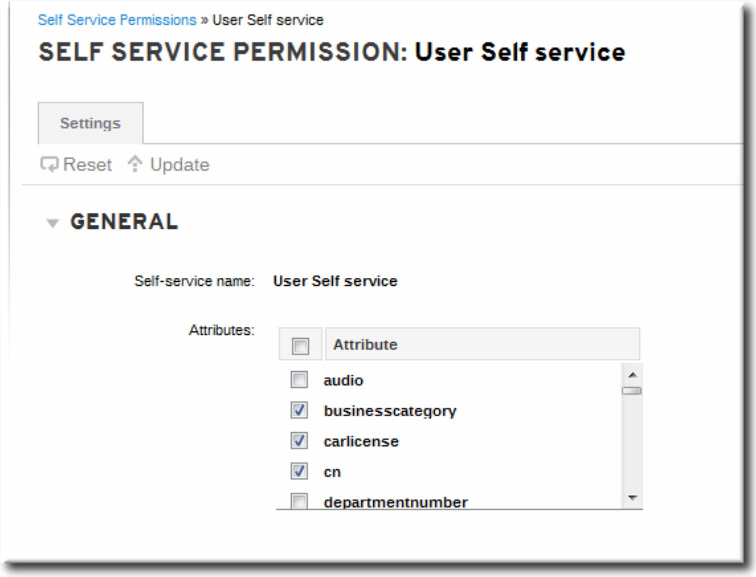


Self-service name: Adding personal info

Attributes:

□	Attribute
☑	initials
□	internationalisdnnumber
□	ipauniqueid
☑	jpegphoto

Enter the name of the rule in the pop-up window. Spaces are allowed:



Self Service Permissions » User Self service

SELF SERVICE PERMISSION: User Self service

Self-service name: User Self service

Attributes:

□	Attribute
□	audio
☑	businesscategory
☑	carlicense
☑	cn
□	departmentnumber

Reference:
https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Identity_Management_Guide/defining-roles.html

LAB 7: IdM Replication

Target server: idm-server.example.com and idm-replica.example.com
Access: ssh root@idm-server.example.com ssh root@idm-replica.example.com

On the idm-replica.example.com run:

```
yum install ipa-server bind-dyndb-ldap
```

On The idm-server.example.com run:

```
ipa-replica-prepare idm-replica.example.com --ip-address 172.16.0.53
```

Copy the replication info to the replica:

```
scp /var/lib/ipa/replica-info-idm-replica.example.com.gpg root@172.16.0.53:/var
```

On idm-replica.example.com run:

```
ipa-replica-install --no-forwarders  --setup-dns
/var/replica-info-idm-replica.example.com.gpg
```

Other options:

```
ipa-replica-install --forwarder=<our forward DNS> --setup-dns <replica file.gpg>
```

Replication verification.

```
ipa-replica-manage list
ipa-replica-conncheck --replica idm-replica.example.com
```

Issue native LDAP quires to check users and notice changes in both servers:

```
ldapsearch -h localhost -Y GSSAPI -b cn=config
"(objectclass=nsds5ReplicationAgreement)"

ldapsearch -h localhost -Y GSSAPI -b cn=config "(nsDS5ReplicaId=*)"

ldapsearch -x -h localhost  -b cn=config -D"cn=Directory Manager"
"(objectclass=nsds5ReplicationAgreement)"  -LLL -W  nsds5replicaLastUpdateStatus

ldapsearch -x -h localhost  -b cn=config -D"cn=Directory Manager"
"(nsDS5ReplicaId=*)"  -LLL -W  nsds5replicaLastUpdateStatus
```

LAB 8: Services and Keytabs

Target server: idm-server.example.com or idm-client.example.com
Access: ssh root@idm-server.example.com ssh root@idm-client.example.com

Log in to idm-access machine:

```
yum install httpd mod_nss mod_wsgi mod_auth_kerb ipa-admintools
```

Prepare content for idm-access:

```
cp workshop.conf /etc/httpd/conf.d/workshop.conf
cp workshop.wsgi /var/www/cgi-bin/workshop.wsgi
chmod +x /var/www/cgi-bin/workshop.wsgi
```

Create the IPA service entry for idm-access:

```
kinit
Password for admin@EXAMPLE.COM:
ipa service-add HTTP/`hostname`
ipa service-show HTTP/`hostname`
```

Retrieve a keytab for httpd service on idm-access:

```
ipa-getkeytab -p HTTP/`hostname` -k http.keytab -s idm-server.example.com
klist -kt http.keytab
```

Configure idm-access to use the keytab:

```
mv http.keytab /etc/httpd/conf/
chown apache:apache /etc/httpd/conf/http.keytab
chmod 0400 /etc/httpd/conf/http.keytab
service httpd restart
```

Access idm-client and run:

```
yum install firefox xorg-x11-xinit.x86_64
exit
ssh root@ idm-client.example.com -X
firefox
```

In Firefox, access idm-access.example.com/test, when you exit Firefox check:

```
klist
```

It might not work as selinux will deny the http-keytab.

```
Cd /root
grep httpd_t  /var/log/audit/audit.log  | audit2allow -m http-keytab >
http-keytab.te
grep httpd_t /var/log/audit/audit.log | audit2allow -M http-keytab
semodule -i http-keytab.pp
```

Now, check again Firefox, after authentication it should print:

```
Hello!
Received connection from 172.16.0.51

YAY! Kerberos authentication works!
Remote user is admin@EXAMPLE.COM
```

To Allow authentication for this small web application without password:

- In the address bar of Firefox, type *'about:config'* to display the list of current configuration options.
- In the Filter field, type negotiate to restrict the list of options.
- Double-click the network.negotiate-auth.trusted-uris entry to display the Enter string value dialog box.
- Enter the name of the domain against which you want to authenticate, for example, example.com.
- Repeat the above procedure for the network.negotiate-auth.delegation-uris entry, using the same domain.
- Restart Firefox, you should see the YAY message with no user-name and password.