Identity Management based on FreeIPA

SLAC 2014

Thorsten Scherf
Red Hat EMEA
What is an Identity Management System (IdM)

- An IdM system is a set of services and rules to manage the users of an organization
- It includes information about individuals, computers, groups, roles, authentication and authorization rules that apply to the set of users and devices managed by the system
- If you need to manage more than a handful of machines you do not want to manually configure all these functions on each one, instead you use an IdM system generally hosted on a centralized server
What is FreeIPA

- IPA stands for Identity, Policy, Audit
  - FreeIPA open source project was started in 2007
  - FreeIPA v1 was released in 2008
  - FreeIPA v3.3 was released in April 2014
- It's based on well known open source tools and standards
- FreeIPA (or just IPA) is the upstream project for Red Hats Identity Management solution
Main values

- Identity and authentication is a complex problem – many disjoint technologies exist
- We want to make it more simple to deploy and use
- IPA is a domain controller for Linux/UNIX environment
  - Think Active Directory but for Linux
  - Central server that stores identity information, policies related to identities and performs authentication
An FreeIPA server is an identity and authentication server. The primary FreeIPA server, essentially a domain controller, uses a Kerberos server and KDC for authentication. An LDAP backend contains all of the domain information, including users, client machines, and domain configuration.
Features

- Centralized authentication via Kerberos or LDAP
- Identity management:
  - users, groups, hosts, host groups, services, netgroups
- Manageability:
  - Simple installation scripts for server and client
  - Rich CLI and web-based user interface
  - Pluggable and extensible framework for UI/CLI
  - Flexible delegation and administrative model
  - Self service portal
Features (Continued)

- X.509 certificate provisioning for hosts and services
- Host-based access control (HBAC)
- Centrally-managed SUDO
- SELinux policy management
- SSH key management
- Group-based password policies
- Can act as NIS server for legacy systems
- Painless password migration
- Integrated DNS server managed by IPA
Features (Continued)

- Replication:
  - Supports multi-server deployment based on the multi-master replication
  - User replication with MS Active Directory
  - Password replication based on passsync.msi
- Cross Kerberos-Realm Trust for IdM <=> AD setups
- Compatibility with broad set of clients
SSSD (System Security Services Services Daemon)

- Retrieves identity information from a central identity management system
- Performs authentication and password change against a central authority
- Enforces access control
- Integrates with client side components like SUDO, SELinux, SSH
- Replaces older technologies including:
  - NIS, direct PAM/NSS LDAP/Kerberos connections, NSCD, winbind
SSSD Architecture

Client

Identity Server

Network Boundary

Authentication Server

 NSS Responder

Identity Provider

Domain Provider

 Auth Provider

PAM Responder

Cache

SSSD
Client - Server Interaction

**SSSD** provides the user authentication for the machine and enforces host-based access control rules

**nss_Idap** fetches object using encrypted LDAP connection

**Certmonger** monitors and renews the certificates on the client, it can request new certificates for the services on the system (NSS and PEM)
IPA and Active Directory

- IPA and Active Directory both provide identity management solutions on top of the Kerberos infrastructure.
- Integration either based on trust or replication.
- IPA AD trust feature is designed:
  - To give Active Directory users access to IPA resources.
  - To allow IPA servers and clients to resolve identities of AD users and groups.
- IPA AD trust feature does not require:
  - Synchronizing accounts and passwords with AD.
  - Installing any software on AD domain controllers.
Cross-realm trust: IdM and Active Directory

• IPA exposes its own realm as an Active Directory-compatible forest

• Two Active Directory-compatible forests can trust each other

• As result:
  • Active Directory users can access IPA resources
  • IPA servers and clients can resolve identities of AD users and groups
  • Access to IPA is controlled by IPA rules (HBAC, ...) for Active Directory users and groups
  • All AD user and group management stays at AD side
AD – IPA replication

A DNS zone is delegated by AD to IdM to manage Linux environment.

Name resolution and service discovery queries are resolved against IdM.

Users are synchronized from AD to IdM.

SSSD
- Authentication
- Identities
- Name resolution

Policies
- sudo
- hbac
- automount
- selinux

Linux System
AD - IPA Trust

Domains trust each other. Users stay where they are, no synchronization needed.

A DNS zone is delegated by AD to IPA to manage Linux systems or IPA has an independent namespace.

Client software connects to the right server depending on the information it needs.
Cross-realm trust: DNS integration

- DNS is the cornerstone for FreeIPA and Windows to discover services in the local and remote domains

- Two configuration options:
  - Conditional forwarder
  - Delegation (recommended)
New AD trust features in FreeIPA-3.3

- Supports Windows Server 2012 R2
- POSIX attributes stored in AD
- Multiple child domains in AD forest
- Legacy clients support for AD integration
- Multiple FreeIPA trusts servers
Host based access control

Use case: Deny all access for everybody, but allow ssh

# ipa hbacrule-del allow_all (also possible during install time)

• Create a new rule idm-users-ssh and assign all hosts

# ipa hbacrule-add --hostcat=all idm-users-ssh

• Add a group to the rule that should get access

# ipa hbacrule-add-user --groups=ipausers idm-users-ssh

• Finally add the ssh service to the rule

# ipa hbacrule-add-service --hbacsvcs=sshd idm-users-ssh
Central sudo Configuration

**Use case:** Sudo user should be able to read system logs

- First create a command-group and add commands to it

  ```
  # ipa sudocmdgroup-add --desc 'log reading cmd' logfiles
  # ipa sudocmd-add --desc 'read logs' '/usr/bin/less /var/log/messages'
  # ipa sudocmdgroup-add-member --sudocmds '/usr/bin/less /var/log/messages' logfiles
  ```
Central sudo Configuration II

- Now create the main sudo rule
  # ipa sudorule-add logfiles-cmd

- Add the command group or single commands to the rule
  # ipa sudorule-add-allow-command --sudocmds
  '/usr/bin/less /var/log/messages' logfiles-cmd

  # ipa sudorule-add-allow-command --sudocmdgroups
  logfiles logfiles-cmd

- Add hosts or hostgroups to the rule
  # ipa sudorule-add-host --hosts tiffy logfiles-cmd

  # ipa sudorule-add-host --hostgroups admin-hosts logfiles-cmd

- Add user or usergroups to the rule
  # ipa sudorule-add-user --user sudouser logfiles-cmd
  # ipa sudorule-add-user --group sudogroup logfiles-cmd
Client sudo Configuration (past)

- Prepare NSS

```bash
# echo "sudoers: sss" >> /etc/nsswitch.conf
```

- Prepare sssd (/etc/sssd/sssd.conf)

```
[sssd]
[...]
services = nss, pam, ssh, pac, sudo
```

```
[domain/idm.coe.muc.redhat.com]
sudo_provider = ldap
ldap_uri = ldap://grobi.idm.coe.muc.redhat.com
ldap_sudo_search_base = ou=sudoers,dc=idm,dc=coe,dc=muc,dc=redhat,dc=com
ldap_sasl_mech = GSSAPI
ldap_sasl_authid = host/tiffy.idm.coe.muc.redhat.com
ldap_sasl_realm = IDM.COE.MUC.REDHAT.COM
krb5_server = grobi.idm.coe.muc.redhat.com
```
Client sudo Configuration (new)

- Now part of regular client setup
- Configures NSS and SSSD

```
# git log ef3c9d3
* ef3c9d3 - (2014-05-09 13:57:04 +0300) ipa-client-install: Configure sudo to use SSSD as data source
```
SELinux user mapping

**Use case:** Every user should get a default SELinux identity

```
# ipa config-show
Maximum username length: 32
Home directory base: /home
Default shell: /bin/bash
Default users group: ipausers
Default e-mail domain: idm.coe.muc.redhat.com
Search time limit: 2
Search size limit: 100
User search fields: uid,givenname,sn,telephonenumber,ou,title
Group search fields: cn,description
Enable migration mode: FALSE
Certificate Subject base: O=IDM.COE.MUC.REDHAT.COM
Password Expiration Notification (days): 4
Password plugin features: AllowNThash
SELinux user map order: guest_u:s0$xguest_u:s0$unconfined_u:s0-s0:s0:s0:c0.c1023$unconfined_u:s0-s0:c0.c1023
Default SELinux user: unconfined_u:s0-s0:c0.c1023
Default PAC types: MS-PAC
```
SELinux custom user mapping

Use case: Every admin user should have staff_u

```
# ipa selinuxusermap-add --selinuxuser=staff_u:s0-s0:c0.c1023 adminrole

# ipa selinuxusermap-add-user --groups=admins adminrole

# ipa selinuxusermap-mod --hostcat=all adminrole
```
SSH-Key management for users

**Use case:** Users have a SSH-Key as part of their LDAP object

```
# ipa user-mod tscherf --sshpubkey="ssh-rsa AAA..
```

-------------
Modified user "tscherf"
-------------
User login: tscherf
First name: Thorsten
Last name: Scherf
Home directory: /home/tscherf
Login shell: /bin/sh
Email address: tscherf@idm.coe.muc.redhat.com
UID: 1094200001
GID: 1094200001
Account disabled: False
**SSH public key:** ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA9IS/LvA5lv7a5wdKLNvLPoDiPU7W1I41Gn3pjobN9zV1tE7zPWj2SKHuV2lXn0u993959nGFn173mQpT5Ct5fe0WPGuAmraegtVCAgfwKQXRHA7RiaQPDkeSVXxAMPrvqPedoeYlt/j9ly+7JahXYcHW3OUR0N0eGFeoIqwg8tX9hr7qRHDQMJrURSnnCT+Pow3P62Hs3x2fbCR4Pdlpeb7Y8woo11TthEjwSHSiKD+qKXT6zu+3dXNftq+dGaahjq3lPfPmgAVyKckO8Puhbb31MzRA3K59LOvyKY5zx8Wg/cpt1rvdvQruFcysU5PFMs6VZYdfwP/Y0KM5jzJvRw==
tscherf@vm236.idm.coe.muc.redhat.com
Password: True
Member of groups: ipausers
Kerberos keys available: True
tscherf@vm236.idm.coe.muc.redhat.com (ssh-rsa)
SSH-Key management for users: SSH-Config

- OpenSSH server config is automatically configured to lookup userkey in LDAP via sssd-Proxy

```
# cat /etc/ssh/sshd_config
AuthorizedKeysCommand /usr/bin/sss_ssh_authorizedkeys
```

- Login using SSH-Keys instead of Kerberos-Principal

```
# ssh -o GSSAPIAuthentication=no tiffy
Mar  8 13:40:13 tiffy sshd[15087]: Accepted publickey for tscherf from 10.32.69.236 port 44882
Mar  8 13:40:13 tiffy sshd[15087]: pam_unix(sshd:session): session opened for user tscherf
```

- Login using Kerberos-Principal instead of SSH-Keys

```
# ssh tiffy
Mar  8 13:38:00 tiffy sshd[15036]: Authorized to tscherf, krb5 principal
tscherf@IDM.CO.E.MUC.REDHAT.COM (krb5_kuserok)
Mar  8 13:38:00 tiffy sshd[15036]: Accepted gssapi-with-mic for tscherf from 10.32.69.236 port 49269
ssh2
SSH-Key management for hosts

- Host keys are automatically added to LDAP during enrollment
- OpenSSH client config is automatically configured to lookup hostkeys in LDAP via sssd-Proxy

```bash
# cat /etc/ssh/ssh_config
GlobalKnownHostsFile /var/lib/sss/pubconf/known_hosts
ProxyCommand /usr/bin/sss_ssh_knownhostsproxy -p %p %h
```

```bash
# ipa host-show grobi.idm.coe.muc.redhat.com
Host name: grobi.idm.coe.muc.redhat.com
[...]
Keytab: True
SSH public key fingerprint:
```
Enable AD trust service on FreeIPA

# ipa-adtrust-install
# wbinfo --online-status

BUILTIN : online
IDM : online

# ipa trust-add --type=ad coe.muc.redhat.com
--admin=Administrator --password

Active directory domain administrator's password:
-----------------------------------------------------------
Added Active Directory trust for realm "coe.muc.redhat.com"
-----------------------------------------------------------
Realm name: coe.muc.redhat.com
Domain NetBIOS name: COE
Domain Security Identifier: S-1-5-21-358654134-3175511377-4185601054
Trust direction: Two-way trust
Trust type: Active Directory domain
Trust status: Established and verified

# wbinfo --online-status

BUILTIN : online
IDM : online
COE : online
Resources

Project wiki: https://www.freeipa.org

Code:
https://git.fedorahosted.org/cgit/freeipa.git/

SSSD:
https://fedorahosted.org/sssd/

Mailinglists:
freeipa-users@redhat.com
freeipa-devel@redhat.com
freeipa-interest@redhat.com