FreeIPA - Control your identity

LinuxAlt 2012
Martin Košek, <mkosek@redhat.com>
Sr. Software Engineer, Red Hat
Nov 3rd, 2012

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License.
1 Introduction
   - Problem definition
   - How to solve it

2 FreeIPA

3 Features
   - Identity Management
   - Policies
   - Client configuration

4 Q&A
The Identity Management problem

- Management of individual identities
  - users, hosts, ...
- Authentication, authorization
  - Policies, ACLs
- Privileges, permissions within or across systems
  - Can be configured on one computer
    - /etc/passwd, SUDO, PAM, ...
- Different interfaces and languages
- May become a synchronization nightmare on network

**centralize!**
Ideal solution

- Central location (but with redundancy!)
- Secure but easy to use
- Based on industry standards
- Single sign-on
- Allow access control (and self-service) on data
- Privilege delegation and separation

Available solutions:

- **NIS, NIS+** - deprecated
- **LDAP** - RFC 4511 (†Kerberos) - current industry standard
  - 389 Directory Server, OpenLDAP, AD
The building blocks

- **LDAP** - data storage
  - Tree-like data structure
  - Good access control granularity (ACI)
  - Optimized for read operations - stale data
  - Multi-master replication
- **Kerberos** - authentication
  - Single sign-on
  - Centralized, KDC knows all the secrets
  - Identity represented by a principal: admin@EXAMPLE.COM
  - Can speak AD language and create two-way trusts
- Additional services: CA, DNS, NTP, ...
- All this can be built manually...

... BUT

- Did you ever try configuring LDAP+Kerberos+other services manually?
- Difficult management for regular admin
The building blocks

- **LDAP** - data storage
  - Tree-like data structure
  - Good access control granularity (ACI)
  - Optimized for read operations - stale data
  - Multi-master replication
- **Kerberos** - authentication
  - Single sign-on
  - Centralized, KDC knows all the secrets
  - Identity represented by a principal: admin@EXAMPLE.COM
  - Can *speak* AD language and create two-way trusts
- Additional services: CA, DNS, NTP, ...
- All this can be built manually...

... **BUT**

- Did you ever try configuring LDAP+Kerberos+other services manually?
- Difficult management for regular admin
FreeIPA

What does IPA stand for?

- Identity - who you are
- Policy - what are you allowed to do
- Audit - who and when accessed

Advantages:

- Easy installation and setup
  - \texttt{ipa-\{server\,replica\,client\}-install}
- Interface to administer identities (users, groups), policies...
  - CLI, WebUI, custom RPC interface
- Linux clients are first-class citizens
  - Native support of Linux services - \texttt{autofs, SUDO, SELinux, ...}
- Redundancy - multi-master replication, read-only replicas, hubs
High-level architecture
Example: add a user

Add a user via LDIF

# ldapadd -D "cn=Directory Manager" -x -W
dn: uid=jdoe,ou=Users,dc=example,dc=com
objectclass: posixAccount
objectclass: person
uid: jdoe
uidNumber: 1001
gidNumber: 1001
sn: Doe
cn: John Doe
userPassword: PAsSw0rd
homeDirectory: /home/jdoe
Example: add a user (cont.)

Add a user with FreeIPA CLI [1/2]: kinit

# kinit admin
Password for admin@EXAMPLE.COM:
# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: admin@EXAMPLE.COM

Valid starting      Expires           Service principal
10/31/12 11:28:02   11/01/12 11:28:01   krbtgt/EXAMPLE.COM@

- Note: ticket lifetime can be configured with Kerberos ticket policy
Example: add a user (cont.)

Add a user with FreeIPA CLI [2/2]: run IPA command

```
# ipa user-add --first=John --last=Doe jdoe --random
--------------
Added user "jdoe"
--------------
User login: jdoe
First name: John
Last name: Doe
Full name: John Doe
Display name: John Doe
Initials: JD
Home directory: /home/jdoe
GECOS field: John Doe
Login shell: /bin/sh
Kerberos principal: jdoe@EXAMPLE.COM
Email address: jdoe@example.com
Random password: +MK2XkIN=vVM
UID: 1998400002
GID: 1998400002
Password: True
Kerberos keys available: True
```
Example: add a user (cont.)
Section 3
Features
Identity Management

- Users, groups:
  - Automatic and unique UIDs
  - Manage users’ SSH public keys, SELinux context
  - Role-based access control, self-service
- Hosts, group of hosts, services:
  - Manage host or service certificates (e.g. secure web server)
  - Automatic group membership for users and hosts

Add new identity object

```bash
# ipa user-add --first=John --last=Doe jdoe --random
# ipa group-add labusers --desc "Lab Users"
# ipa group-add-member labusers --users=jdoe

# getent passwd jdoe
jdoe:*:94800185:94800185:John Doe:/home/jdoe:/bin/sh
# getent group labusers
labusers:*:94800186:jdoe
```
Identity Management (cont.)

- Cooperation with Active Directory domains
  - Till 3.0: winsync+passsync - synchronize AD users to FreeIPA
  - From 3.0: Cross-realm Kerberos trust
- FreeIPA + AD domain with a Trust is recommended way to manage Windows and Linux hosts

Create Active Directory trust

```
# ipa trust-add --type=ad ad.domain \ 
   --admin Administrator --password
```

- Linux hosts now accessible with GSSAPI-aware Windows SSH client
DNS

- Add new A, AAAA, CNAME, ... records with IPA interface
- Controlled with `bind-dyndb-ldap` plugin
  - Provisions BIND with records from LDAP

Add new DNS records

```bash
# ipa dnszone-add lab.example.com \
   --name-server=ipa.example.com
# ipa dnsrecord-add lab.example.com pc01 \
   --a-rec=10.0.10.1 --a-create-reverse
```

- Updated automatically with client install or IP address change (by SSSD)
Policy - HBAC

- Control who can do what with **Host Based Access Control**
- Enforced by **SSSD** for authentication requests via **PAM**

HBAC - rule example

# ipa hbacrule-show labmachines_login
Rule name: labmachines_login
Source host category: all
Enabled: TRUE
User Groups: labusers
Host Groups: labmachines
Services: sshd, login
Policy - Other services

**SUDO - rule example**

```
# ipa sudorule-show labadmin_yum
  Rule name: labadmin_yum
  Enabled: TRUE
  RunAs User category: all
  RunAs Group category: all
  User Groups: labadmins
  Host Groups: labmachines
  Sudo Allow Commands: /usr/bin/yum
```

- **Automount** - automatic NFS mounts
- **SELinux rule** - similar pattern to HBAC, assign SELinux user context per-host
And now for the client part...

- It is nice to have a server, but client configuration matters too
- There is a lot to configure - users, auth, services...
- IPA client installer should make it easier:
  - Configures SSSD - our client project
  - Synchronizes time with IPA server via NTP (Kerberos!)
  - (Optional) Upload public SSH key of the host
  - (Optional) Creates DNS record for the client in IPA

Prepare client record (optional)

```
# ipa host-add client.example.com --random
------------------------------------------Added host "client.example.com"
------------------------------------------
  Host name: client.example.com
  Random password: 7QGk+eHU.Y8U
  Password: True
  Keytab: False
  Managed by: ipa.example.com
```
And now for the client part... (cont.)

Configure an IPA client

```
# ipa-client-install --password 7QGk+eHU.Y8U \  
   --unattended
Discovered IPA realm EXAMPLE.COM
Hostname: client.example.com
Realm: EXAMPLE.COM
DNS Domain: example.com
IPA Server: ipa.example.com
BaseDN: dc=example,dc=com
Synchronizing time with KDC...
Enrolled in IPA realm EXAMPLE.COM
...
DNS server record set to: client.example.com -> 10.0.0.10
...
Configured /etc/ssh/ssh_config
Configured /etc/ssh/sshd_config
Client configuration complete.
```
Section 4
Q&A
Resources, contact

- Web + wiki: www.freeipa.org
- Code: www.fedorahosted.org/freeipa/
- IRC: #freeipa on freenode
- Mailing lists:
  - freeipa-interest@redhat.com
  - freeipa-users@redhat.com
  - freeipa-devel@redhat.com

Questions?
The end.

Thanks for listening.